

一、前言

本研究報告乃「建立大地工程鑽探試驗資料庫系統」之系列報告之一。本研究案包含：

- (一)大地工程鑽探、試驗資料庫來源調查與全程規劃。
- (二)大地工程鑽探、試驗資料庫法規草案之調查與建議。
- (三)台北市、高雄市兩都會區之實質建檔。
- (四)土層剖面圖專家系統設計之研究。

研究方案(一)乃由淡江大學土木工程系副教授倪至寬博士共同主持，針對全國性之縣市建管單位，公私工程單位和顧問公司之大地工程(鑽探、試驗)資料完成訪詢調查工作，以確實掌握資料來源及數量。研究方案(二)由經濟部中央地質調查所副所長簡芳欽先生共同主持，邀請法學，地質和大地工程等方面專長共同草擬土地工程資料報告之標準格式、歸屬權和管理辦法等制度。研究方案(三)由計劃主持人國立台灣工業技術學院營建工程技術系李咸亨博士與國立成功大學土木工程系陳景文博士共同負責台北市和高雄市兩都會區之實際建檔工作。第一年各以 500孔為目標，測試本系統之格式及大量建檔時之實用性，以便未來交由各地方使用單位時無系統誤差發生。研究方案(四)則由國立台灣工業技術學院工業管理技術系林玉斌副教授共同負責研究開發「土層剖面專家系統」，以求達到地層分類自動化之目的。

大地工程地質資料庫則可利用查詢之方式，得到某些相鄰鑽孔之資料，並進而由人員研判這些鑽孔所含蓋範圍的土層地質分佈狀況。

但是地層的研判，必須由有相關知識與經驗的專家方能勝任，因此本研究即考慮到以現今非常盛行的專家系統技術與已開發完成的大地工程地質資料庫相結合，以協助地質學專家或營建設計工程師從事土層研討或工程規劃的參考。

本研究之主要目的是要以經驗法則找尋出較適當的土層匹配方式，並以此法則為基礎，建構一土層匹配之推理機構。

本研究最後希望能產生一土層剖面分析之專家系統，此系統可依據目前部份的大地工程地質鑽探資料，推論出較全面的地層分佈，並以圖示的方式，使得資訊表現得更為具體，且較能讓使用者接受。

土層剖面圖之分層，主要是依據大地工程地質資料庫之鑽探資料，由專家系統推論各鑽孔本身與其他鑽孔之間資料的相關性而予以分層。而影響土層匹配的資料相當多，如土層之深度、土／岩分類、N 值、各種土／岩的成份比例、自然含水量、液性限度、塑性指數、與孔隙比等。

在專家系統建構的初期，不宜考慮如此多的因素，因此本研究計畫先考慮土層深度，土／岩分類，與 N 值等基本資料，作為土層分層之依據。並且本研究也假定所參考的鑽孔內各土層的土／岩分類是正確無誤，其與其他的物性資料不相違悖。

二、專家系統

今日專家系統 (Expert System) 之應用，已是人工智慧 (Artificial Intelligence) 研究多年來發展較為成熟，成效也較為顯著的一門領域。本章將對專家系統作一般性地介紹，並進而描述其組成構造。

2.1 專家系統之定義

專家系統係利用人工智慧中“問題解決 (Problem Solving) 技術”，配合某特殊領域中大量的知識，形成一種推論程序，以解決該特殊領域中的問題。

史丹福大學的Feigenbaum教授對專家系統下的定義是：專家系統是一有智慧的電腦程式，它使用知識與推理程序去解決問題，而這些問題困難到必須靠專家的經驗方能解決 [5]。也就是說，專家系統是一模倣人類專家決策過程之計算機系統。

專家系統可用特定的硬體與軟體來設計，也可以是使用於一般目的之電腦的軟體系統來設計。

專家系統的知識庫是由特定範圍中各種行為的法則或經驗等資訊所構成。通常法則可宣告出各種狀況之因果關係，而經驗資訊則是由特定領域中人類專家的經驗搜集而得，因此專家系統均限於某一特定領域。

從專家系統的應用範圍來看，專家系統是一計算機程式，它能提供建議、分析、分類、溝通、諮詢、設計、診斷、說明、勘查、預測、辨認、概念形成、解釋、證明、學習、管理、監督、規劃、排程、檢索、測試及教學等功能，而這些功能均須人類專家的協助方能完成。可見專家系統的應用範圍相當廣泛。[16]

在不同領域的應用裡，它可能配合使用環境的性質與需求，來扮演專家的角色；也可能只是專家的助手，或人類工作上的夥伴或顧問，也可能是做為訓練用的教師。

在一般問題解決的過程中，知識是解決問題成功與否的關鍵之一。例如，一位地質學者能夠有效地探測出地層是否有礦藏，主要是依據本身的學問與經驗以解決其面對的狀況。專家的知識可說是理論與一些問題解決的經驗與法則累積而成的。專家系統即是利用這些人類專家的知識，以計算機能執行的推理機構來建造。

因此專家系統所欲達成目標應有下列特性 [15]：

1. 代替一稀有的人類專家。
2. 融合多位人類專家之知識與經驗。
3. 用來訓練或教導新的專家。
4. 當無法獲得人類專家支援時，專家系統可提供專家知識之查詢。
5. 若專案無法留住專家時，專家系統可提供必要的專家知識。

受限於不同知識領域，知識表達方式的差異性，與推理過程的特殊性，以現今可用的技術，只能在於較窄的應用領域裡，建立專家系統。

2.2 專家系統之分類

至今專家系統已建構於問題解決的領域包括醫藥、數學、工程、化學、計算機科學、法律、國防、農業、設備維護、財務、管理、製造、運輸、辦公室自動化、教育、訓練等 [9]。

將專家系統的功能加以分類，其含蓋的功能有：[27]

1. 解釋 (Interpretation)

從觀察的資料中，如影像、電子波、光波等，形成高層次有用的結論，如PROSPECTOR, DENDRAL。

2. 預報 (Predicate)

由給定的狀況，預測未來發展，如氣象預報。

3. 診斷 (Diagnosis)

根據一些可發現的徵狀為基礎，在複雜的情況之下，找出故障的原因。例如，根據病人的症候診斷出其疾病。此類系統有 MYCIN，PUFF 等。

4. 設計 (Design)

在滿足一些給定的限制之下，決定一系統的架構，以滿足一既定目標。例如，建築設計。

5. 計劃 (Planning)

在既定的起始狀況之下，產生一系列的活動，以達到某目標。如財務計劃、作戰計劃。這類系統有 TATR，MOLGEN 等。

6. 監視 (Monitoring)

以預期的行為作準則，比較現行系統的行為。

7. 除錯與恢復 (Debugging and Repair)

找出錯誤之所在並解決之。如程式偵錯，DART 即屬這一類。

8. 教育 (Education)

訓練學習者了解某領域的知識，予以檢查並更正。

9. 控制 (Control)

主導一環境的行為。

2.3 專家系統之架構

專家系統已被應用於許多領域，而不同的專家系統，組成架構也不盡相同，但無論如何，大部份的專家系統通常都有一些共同的部份：使用者界面 (User Interface)，知識庫 (Knowledge Base)，推論機 (Inference Engine) 又稱推理機，解釋

界面(Explanation Facility)及知識擷取界面 (Knowledge Acquisition Facility) 等。如圖2.1所示。

使用者界面允許使用者在特定狀況下，詢問某領域相關之問題，並輸入一些狀況，做為專家系統推論之依據，並將其推論之結果反應給使用者。

知識庫可被分成知識資料庫 (Knowledge Database)及領域資料庫 (Domain Database)。知識資料庫包含此領域內之規則 (Rule)，這些規則係以某一特定格式表示，如“若有一資產，其在一年內是易動的，則其在會計項目中，被視為流動資產”。領域資料庫則包括該領域之事實 (Facts)，領域資料庫也有的格式表現其內容，例如“現金是流動資產”。

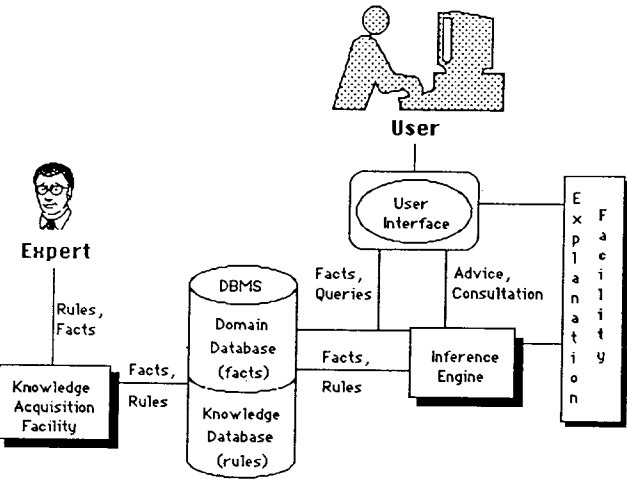


圖 2.1 專家系統架構圖 [15]

推理機模擬該領域之人類專家的思考方式，並使用資料庫所提供之資訊與使用者輸入之詢問狀況，以推論出一事實結果，此事實結果即為使用者詢問之解答。

解釋界面會紀錄其推理過程中各階段的結果及其所使用過程的法則與路徑，以便快速且以既定的格式告知使用者其所獲得之結果的依據。

知識擷取界面是人類專家與知識庫交談的工具，其目的是幫助將專家的經驗與知識用規則或事實的形式貯存於知識庫內之知識資料庫或領域資料庫，或可利用知識擷取界面查詢及更改知識庫內之內容。

通常專家所獲得之知識與經驗都是由學習而來，因此為了讓專家系統也有學習的能力，有些專家系統更增加了自我學習(Self-training)的功能，圖2.2即為此類專家系統之架構。

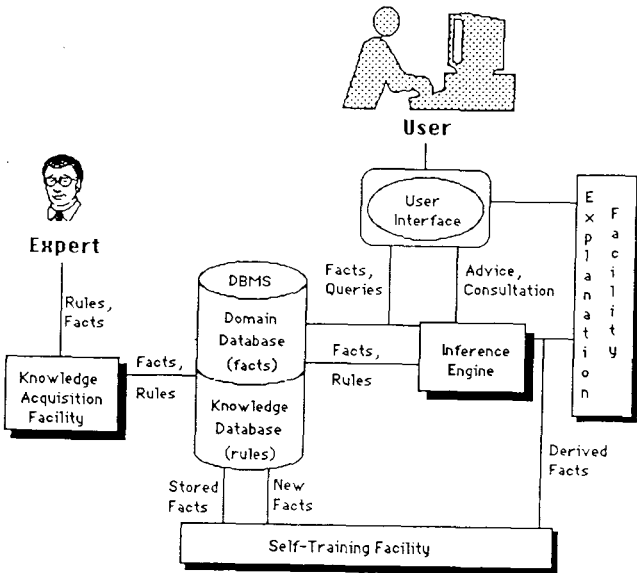


圖 2.2 增加自我學習之專家系統架構圖 [15]

自我學習界面能接受專家系統之推論機推導出的事實，並將這些事實與貯存於知識庫之事實比較，若新推論之事實不存在於既有的知識庫，則將此新推論之事實加於知識庫之領域資料庫內。自我學習界面亦須確定推論機產生之新事實不會與原存於知識庫的事實相衝突。

2.4 知識表示

專家系統是一種容納著知識的電腦程式，它的運作能力是否能達到專家的水準，則端看其包含知識之多寡及有效與否。

因此為了使專家的知識與經驗能夠正確而有效地表示在專家系統之中，知識的表示方式就成為知識庫系統主要研究方向。

知識庫中貯存專家知識的方式，可概分為靜態(Static)與動態(Dynamic)兩種類型。動態的知識庫是一個工作記憶區(Working Memory)，係存放系統推理過程所產生的一些新事實，這些事實可能作為往後推理過程的依據，也可能就是推理最後的結果。靜態的知識庫主要貯存專家的知識與經驗。

知識庫系統發展近30年來，許多的知識表示法已被提出來，Mylopoulos與Levesque兩位曾將這些表示法分成四類[17]：

1. 邏輯表示綱目(Logical Representation Schemes)

這類的表示法係使用邏輯語法表示知識庫內之知識與經驗。

2. 程序表示綱目(Procedure Representation Schemes)

程序表示法係將問題解決的所有步驟對映成程式的指令，而將一連串的指令集合起來，表示成專家系統問題解決的知識。產出系統(ProductionSystem)即是程序表示法之一例。

3. 網路表示綱目 (Network Representation Schemes)

網路表示法是將知識用圖形 (Graph) 的方式表示，其中節點 (Node) 部份表示問題領域內之一標的物 (Object) 或概念 (Concept)，弧線 (Arc) 則表示標的物或概念之間的關係。語意網路 (Semantic Network) 即為一例。

4. 結構化表示綱目 (Structured Representation Schemes)

結構表示法擴充了網路的表示型態，它允許網路內之節點用更複雜的資料型態表示。常見的結構表示有框架 (Frame)

。

上述之各種知識表示法均各有其優點。如產出規則對於程序性的知識表示特別有用；語意網路有利於表達標的物間之關係；以框架為基礎之語意網路，更能貯存大量有關標的物性質與關係的知識；一階邏輯則提供了一種描述較為廣泛而不特定的知識。在現今的專家系統開發上，均趨向各種知識表示法的混合使用。一系統可能用產出規則定出一些程序，用以存取各種事實，而以語意網路描述事實間之關係，並用框架貯存事實內之屬性值。

一個結構良好的知識庫，可較清楚地表示該領域的知識，並且能夠顯著地提高測試與維護的效率。Ken Pedersen提出了一些條例，作為產生結構良好的知識庫的導引 [19]。

1. 將知識簡單化。
2. 將知識與處理知識的控制程序分離開來。
3. 知識中的規則描述，儘量少用“或者 (Else)”。

2.5 推理機之控制策略

專家系統在找尋知識庫的知識時，所引用的方法即其推理機之控制策略，其中包括規則搜尋的方向及搜尋的方法。

專家系統所應用的控制策略，包括後向鏈結 (Backward chaining)，前向鏈結(Forward chaining)，各種的搜尋策略，問題歸納 (Problem reduction)，樣式比對程序(Pattern matching procedure)，單一化(Unification)，事件導向控制 (even_driven control)，及混合(hybride)運用[15]。

以上的各種策略，可能以一種 或多種 混合設置於每個專家系統的推論機中。

以後向鏈結 (Backward Chaining) 與前向鏈結 (Forward Chaining)而言。後向鏈結 又稱目標驅動(Goal-Driven)。後向鏈結是由一假設的結果(目標)開始尋找一些規則或事實，以支持其假設，若沒有可供支持的知識可利用，則更換另一假設，重新至知識庫比對，直到沒有假設產生，然後推理機會找出其支持程度最高的假設做為其推論的結果。

前向鏈結又稱資料驅動 (Data Driven)。它集合一些現實資料做為起始狀況，系統再依序考慮這些狀況的每一項，經推理以達成某一目標。

另一種常用的控制策略是結合目標驅動與資料驅動，先由資料驅動推論出若干的假想結論，然後系統純粹按照目標驅動的方式，依次考慮每一假想結論，但在後向推理過程中，每當一有新證據發現時，系統則轉成前向推理，視其是否有新結論產生出來，或有捷徑可達到目前目標。

推理機也運用不同的搜尋策略找尋知識庫內適當的知識，如先寬度搜尋與先深度搜尋。先深度搜尋在既定目標之下，先針對一個搜尋路徑，一直往下階層搜尋，如圖2.3的 A, B, D直到搜尋至最低之階層，而得到確切的事實(Fact)或搜尋失敗而回溯(back-tracking)到上一階層，再往其他路徑搜尋。

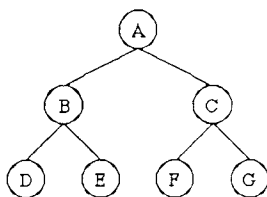


圖 2.3 樹狀搜尋路徑圖

先寬度搜尋時，會先搜尋同一階層的所有規則，而後再搜尋至下一階層，以圖2.3 為例，其搜尋路徑將是A, B, C, D, E, ...。

不同的推理機將依其不同的特性，而引用不同的搜尋策略或混合應用，也可加上經驗法則 (Heuristic)，而使得搜尋的步驟能夠減化。

問題歸納企圖以一結構的方式表示問題。其主要技巧是將問題分解成許多的小部份，而將這些小部份依其關係連接起來，而使得問題推導時較為簡單。

樣式比對之策略可配合後向鏈結使用，當上階層的規則欲起動下階層適當的規則時，即可引用樣式比對之策略。所謂的樣式 (Pattern) 即相當於規則中的前提或結果。

單一化係用於比較不同的串列 (List) 項目，並找到串列中所需要的某一項目，Turbo Prolog 即利用單一化處理串列。

事件導向控制是一種時序性的控制策略，其適合用於程序控制，事件或動作的執行皆有一定的程序，上一步驟完成了，下一步驟才得以開始；上一步驟若無法滿足現況，則下一步驟則無法起動。

2.6 專家系統的生命週期

專家系統發展之前，首先須考慮專家系統建構的可能性，其中須考慮的因素如圖2.4 所示。

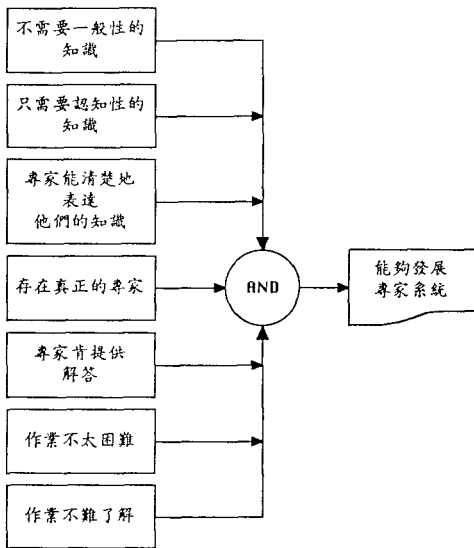


圖 2.4 發展專家系統之基本需求 [27]

若此系統能夠符合發展的條件，則考慮其是否有必要發展

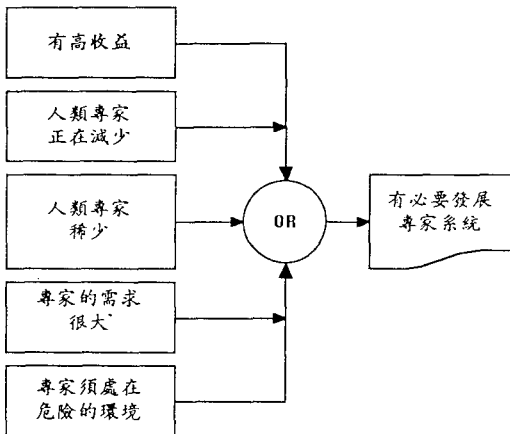


圖 2.5 發展專家系統之必要條件 [27]

然後考慮此問題領域，其特性是否適合用專家系統的方法建構，其考慮因素有圖2.6 所表示之因素。

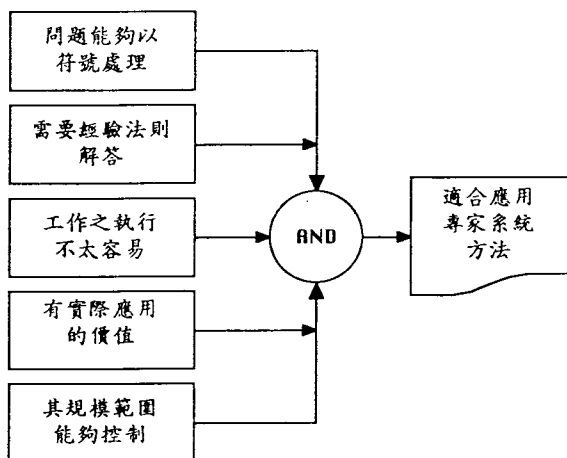


圖 2.6 適合發展專家系統之特性[27]

一旦此專家系統決定發展後，則專家必須盡可能地提供此方面的知識，系統建構者則將這些知識轉換為一結構性的格式，以便及早決定系統建置的方法與程序。

一個專家系統開發之生命週期，描述如下：[15]

A. 系統概念化(Conceptualization) 及需求分析

1. 經由一些問題解決之實作，了解將開發之系統的運作情形，而對此系統形成概念。
2. 系統可行性分析。
3. 預先搜集需求。
4. 預先分析需求。

B. 工具選擇

5. 在既有的需求條件下，考慮所有可用的專家系統發展工具。
6. 將符合本計畫的工具挑出來，並教導系統發展之成員此工具之使用與其功能。

C. 雛型設計 (Prototyping)

7. 發展系統之雛型。
8. 製作雛型之文件。
9. 訓練使用者如何使用此雛型。
10. 讓使用者有機會使用此雛型系統，系統發展者可趁此機會紀錄下使用者使用後之感覺與建議。
11. 發展者視使用者之滿意程度，決定是修改雛型，或接受此雛型為系統之最終產品。

D. 最終製造 (The End Producer)

12. 最終產品之包裝。
13. 最終產品之使用者訓練。
14. 測試使用者接受與否。
15. 若尚無法接受，則重新進入雛型設計階段。
16. 正式操作與維護。

三、土層剖面圖

土層剖面圖係利用地下探勘所獲得的資料描繪而成，其有助於工程的規劃及結構物基礎與其他地內設施的設計，而且尚可幫助選擇適當的施工方式。

在現行的任何大工程，都須作現場地下鑽探以了解地下土壤的詳細數據。例如，設計基礎及施工規劃必須的土壤種類與工程性質等。若須了解地內土壤的物理性質，則可在適當深度採取土樣作試驗室試驗或在現場作試驗求得。

3.1 鑽探計劃

地下探查之前，必須考慮該地區地質條件的複雜程度以及探查時應該解決的問題，以制定一套合理且合乎經濟的探查計劃。[11,14]

探查計劃中，鑽孔的佈置及數量，一般係根據計劃結構物的規模、重要性、配置以及工程地點的地質條件加以擬定鑽孔位置，作有規則且直線的排列。原則上，地質較複雜的區域，需要較密的多數鑽孔；地質單純均一的地層，則少數鑽孔即能獲得令人滿意的地質構造。

鑽孔的方法很多，適當方法的選擇，須依所欲得知的資料種類與範圍，調查區域的土壤情況所需的時間及經費等因素而定。一般較深的土壤調查，都使用機械操作的鑽深器具。有兩種比較常用的方法為：螺旋鑽法 (auger method)，與沖洗鑽法 (wash boring method)。

一般螺旋鑽法使用連續多段鑽頭，則土樣即可順著鑽頭之螺紋傳送至地面。

沖洗鑽法係用壓力將水流導入鑽桿，當鑽桿底部的衝頭(chopping bit)上下反復衝擊及旋轉下面土壤時，加壓水流也同時經由衝頭噴射土壤，衝頭前端的土壤即變軟和水混在一起往上冒。用沖洗鑽法沖洗到地面的碎土，可藉此判別概略的土壤種類與土層變化。

若為求瞭解地內土壤的確實情形，一般都以不經擾亂之土壤作樣本。採取此種土樣必須使用特別的技術與器材於自然土層中鑽取。通常先用沖洗鑽法或螺旋鑽法鑽孔到某一深度，然後將土壤取樣管插入不擾亂的土層中採取土壤試樣。

3.2 由地質探查資料繪製地質剖面圖

由地質探查所獲得的資料，如土壤、岩石之分類、層厚、層深等等之變化情形可作成鑽孔柱狀圖。[11, 14]

在地資複雜以及沉積環境不規則地區，每一鑽孔之柱狀圖，僅能顯示鑽孔周圍附近的地質情形，若欲表現計劃地區整體之地質構造，則應以地層剖面圖為之，亦即依相鄰鑽孔的直線位置及各鑽孔的柱狀地質資料加以推測各孔間相同岩層可能連續延伸的情形。如圖3.1。

在分析探查結果時，地層師應在他提出的剖面圖內說明所得結論的可靠程度。如土/岩層的劃分能確定者，則用實線劃分；若係根據推測所得，則用虛線表示。

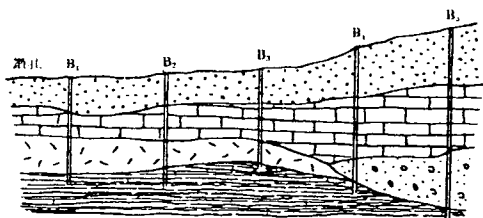


圖3.1 土層剖面圖範例[33]

3.3 土壤分類

大致來說，任何土壤都是由固態顆粒參雜在顆粒間的孔隙組合而成。固態顆粒乃是由不同礦物的小顆粒所造成，孔隙之中則可填充水，空氣或是部份水與部份空氣。就工程的觀點，土壤是由岩床上礦物及有機質等鬆散的聚集物組成的[11]。

土壤的構造即代表其外觀或“感覺”(Feel)，它決定於顆粒的相對大小和形狀，亦決定於其粒徑分佈。因此像砂和礫石等粗粒土壤就有較粗的構造，而細粒土壤則含有絕大多數肉眼所無法看到的微細礦物顆粒。沈泥和黏土即是細粒土壤的最好代表。

依構造來分，土壤可分為粗粒土壤與細粒土壤。傳統上，以肉眼所能看到的最小粒徑做為這兩類土壤的分界線。今日，在土木工程實務上，常採用統一土壤分類系統，以便於對土壤的工程性質與土層的物性指數及分類作有系統的描述。

此統一土壤分類系統係由卡氏 [3] 提出的“統一土壤分類”之雛型衍生而來。本研究所參考的“統一土壤分類新表”(附錄五)，則是李咸亨教授以此系統為基礎而改進的。

統一土壤分類系統的四大區分，如表所示。它們是1.粗粒土壤(coarse-grained)，2.細粒土壤(fine-grained)，3.有機土壤(organic soil)，4.泥炭土(peat)。

粗粒土壤可細分為礫石(G, Gravels and Gravelly soil)與砂土(S, Sands and Sandy soil)。礫石指留於4號篩粗粒比例有較大百分比者。而砂土則指通過 4號篩的部份比例較多者。礫土(G)與砂土(S)二者均被再細分成 4個子目，依粒徑分佈及顆粒性質分GW及SW、GP及SP、GM及SM、GC及SC。GW與SW均為級配良好的礫石及砂土，且通過200號篩的少於50%；GP與SP則為級配不良的礫石與砂土，且含有很少的非塑性細粒或不含。所謂級配良好(W, Well-grained)的土壤係指含有各種大小的顆粒，反之則為級配不良(P, Poorly-grained)的土壤。另GM及SM則為沈泥性(M表沈泥)礫石及砂土；GC及SC表黏土性(C表黏土)礫石及砂土。

通過200號篩超過50%的細粒土壤，可細分為沈泥 (M)及黏土(C)，及依其液限及塑性指數而定。有機土(O)泥炭土(Pt)雖亦可歸類為細粒土，但粒徑大小並沒有指明。沈泥與黏土兩者均再分成3 個子目，分別為：ML是無機性沈泥並含極細砂土，岩石碎屑及沈泥性或黏土性細砂或低塑性之黏土性沈泥；CL是低至中塑性的無機性沈泥，礫石性黏土，沈泥性黏土，貧性黏土；OL是低塑性的有機性沈泥及有機沈泥性黏土；MH是無機沈泥，雲母之細砂性或沈泥性土壤，彈性沈泥；CH是高塑性無機性黏土，肥沃黏土；OH是中至高塑性的有機性黏土及有機性沈泥；Pt則係泥炭土及其他高度有機質土壤。

土壤分類系統可視為大地工程之工程師們相互溝通之語言。本研究所以依據的大地工程資料庫，即以土壤分類系統對土壤的描述為原則予以描述。

表 3.1 土壤分類彙總 [30]

主要區分	次要區分	分類符號
粗 顆 粒 土 壤	礫	GW
		GW-GM
		GW-GM-GC
		GW-GC
		GP
		GP-GM
		GP-GM-GC
		GP-GC
		GM
		GM-GC
	石	GC
		SW
		SW-SM
		SW-SM-SC
		SW-SC
		SP
		SP-SM
		SP-SM-SC
		SP-SC
		SM
土	SM-SC	
	SC	
	ML	
	CL-ML	
	CL	
	OL	
	MH	
	CH	
	OH	
	Pt	

3.4 探勘記錄解釋之專家系統

探勘資料之解釋不僅需要經驗，也需一些物理方面的理論基礎及地理的資料才能預測出地理的狀況。而解釋的模組也需樣式辨識(Pattern Recognition)的技巧，予以支持。因此根據英國Zivy的看法，探勘資料之解釋，至今仍是一門內部爭議頗多且規則多變之學術領域，其解釋模式之設計乃為結合計算機程式和試驗成一整體之高度實驗性之過程。因此，以傳統程式設計方式已無法發展出合適之資料解釋系統，唯有目前專家系統之技術才可能提供令人滿意之解釋能力。[29]

在整個探勘資料的解釋系統的建立過程中，解釋模組的設計，可謂是最重要的部份。解釋模組大多是由一些經驗程序累積而成，它係模仿人類專家的資料解釋程序設計而成。

很大的變異性是探勘資料解釋系統之特性，因為其解釋模組常會因地理區域之不同，地理環境之不同，及許多相關因素，而產生不同的模組。由於這個原因，解釋模組經常會因新理論的提出而有所改變。也因此，一個鑽探解釋軟體的需求與規格之獲得並非易事，所以為獲得一能夠解釋鑽探資料之系統，大多以發展原形(Prototype)的專家系統為起點，而後再因需求之改變再更改原形。

以Dipmeter Adviser系統為例，它是由Schlumberger公司經由4年的努力，利用專家系統的技巧，製作出的油井鑽探資料解釋知識庫系統 [24,25]。自從Schlumberger公司的研究部門引進人工智慧之後，便開始不遺餘力地專注於研究探勘資料解釋的專家系統，其最典型的系統即是DipmeterAdvisor 專家系統。

Dipmeter Adviser系統是利用儀器深入油井鑽孔內部，以測量鑽孔四週表面的一些物理性質，如岩石導電性、傾斜角等。而後便能利用這些寶貴的資料研判是否有礦藏（主要針對碳氫化合物），並設計出鑽探出這些礦藏的方法。

此系統主要分有四大部份：一些產出規則，一個前向鏈結的推理機，一個檢視資料正確性的程序，及一個螢幕驅動的使用者界面。其產出規則是依不同的功能區分成11個群組，在該系統內使用了90條產出規則，而以約30個述詞(Predicate)與函數表示。

另有一著名的有關地質學的 PROSPECTOR 系統，是由史丹福研究機構發展出的專家系統[10]。PROSPECTOR系統是用來輔助地質探勘專家找尋礦脈之用。其運作是由地質探勘專家告訴一些特別重要的探勘特性開始，這些重要特性如地質設定、結構化控制、岩石種類、礦物種類及替代物品的提出或猜測。程式將所得的觀察結果與不同種類的礦脈模型比較，注意其相以點，相異點和缺漏的資訊，然後程式以對話方式要求地質學家收集額外適當的資訊，並使用這些資訊對探勘礦藏做一評估。

PROSPECTOR系統擁有1000條以上的法則，並且使用超過1000個實體(Entity)的地質名詞分類。

到目前為止，另有一些相關於油井探勘及地質研究領域的專家系統也已開始投入於實際的應用。如 CEMENTINGEXP SYS, CONE, Drilling Advisor, PILE, SOILCONE, SOURCE ROCK ADVISOR及WAVES等系統。

CEMENT EXP SYS 是用於油井鑽探時，建議水泥黏合(Cementing)的程序。[9]

CONE是一個以OPS-5 語言寫成的專家系統，主要用途是利用圓錐貫入試驗的結果進行土壤分類，並估算土壤的強度參數。CONE在執行時是先收集資料，再檢核數據的正確性並進行必要的前處理，然後再進行土壤分類和土層剖面圖的工作，最後再來推算土層的相關性質。[18]

Drilling Advisor系統係用於協助油井鑽探監督者，解決油井鑽探時可能發生的問題，包括油井鑽探的技術、設備及程序等問題。[9]

PILE是用LISP語言寫成，用於幫助使用者來選擇基樁的種類。[22]

SOILCON 根據結構物的特性以及已有的土層資料，來規劃需要進行的工址調查。這個系統包含了24種進行工址調查技術的各項特質，這些技術包含了最簡單的地形圖查閱，一直到複雜的現場試驗。[26]

SOURCE ROCK ADVISOR 是為了評估岩石之石油含量而設計的。[9]

WAVES 則是用於幫助發展處理有關地震資料的控制策略。
[9]

四、土層剖面圖專家系統推論機之設計

本文所討論之系統主要功能是使用知識庫內之專家經驗來處理同一孔之土壤分層和研判孔與孔間之連結分割問題。為了達成目的，同一孔在分層時，必須同時考慮到鄰近孔之土質特性與匹配狀況，以決定土壤之分層型態，因此，本研究以一實驗性之層次間相似值計算模式為基礎，以建構較佳相似性之推理機，同時，本研究亦顧及到「合理的估算劃分時間」，以免因為專家系統程式之運算效率令人無法接受。

由於土層之匹配已非一般之推理策略所能達成，本研究即綜合一些控制策略，而改寫一適合土層匹配之推理機構，外建於Turbo Prolog之上。

4.1 推理機設計策略之研究

土層剖面判斷之推理機構，主要在考慮一鑽孔土層與其環境中相鄰鑽孔土層之匹配問題。雖然根據鑽孔取得之土層資料是一有序資料列 (Sequential data)，其匹配之狀況是一種非多項式困難問題 (N - P hard problem)，即當鑽孔內之土層增多時，使用兩鑽孔之土層資料匹配出各種剖面土層的可能狀況，其難度並非以倍數或多項式倍數之型態增加，而是以指數函數成長之方式來增加。簡而言之，當相鄰兩鑽孔各有一層土層資料時，土層之可能狀況有三種，但若各有十層土層資料時，土層的可能狀況，卻高達八百多萬種。

分析土層資料和土層狀況之間的關係，主要是用以決定土層劃分之推理策略。以相鄰兩孔各有一土層資料為例，土層劃分之可能狀況有圖4.1之三種可能性：

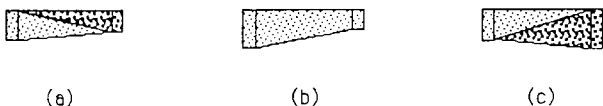


圖 4.1 相鄰兩孔各有一層資料時，土層匹配之三種可能性。

若相鄰兩孔，首孔有二層資料，次孔有一層資料，土層劃分之可能狀況有圖4.2 之五種可能性。

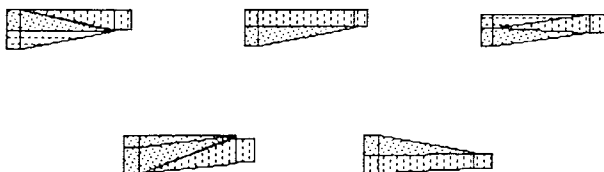


圖 4.2 首孔含二層資料，次孔含一層資料時，相鄰兩孔之五種匹配狀況。

以一鑽孔含 n 個土層分類資料，相鄰鑽孔含 m 個土層分類資料而言，其各種可能之分配函數 $f(n, m)$ 為一種遞迴函數 (recursive function)。由於鑽孔之各土層資料具有自然之次序性， $f(n, m)$ 之定義可證明如下：

(I) 假設兩相鄰鑽孔中，第一鑽孔之土層分有 n 層，第二鑽孔之土層只有一層，則二鑽孔之土層匹配方式，有以下兩種狀況：

狀況 1：第二鑽孔之土層，能夠與第一鑽孔之 n 層土層中之某一土層匹配之可能性有 n 種，如圖4.3所示。

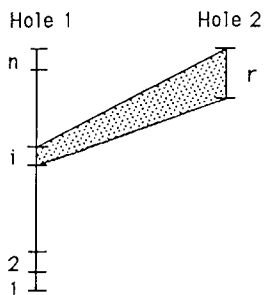
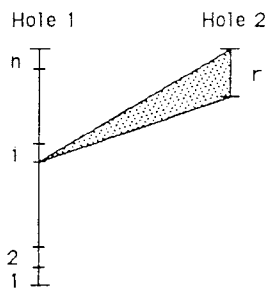


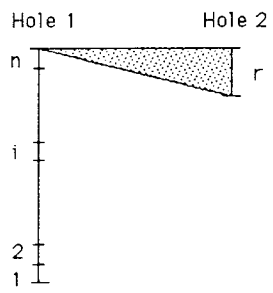
圖 4.3 首孔有 n 層土層，次孔只有一層土層時，
相鄰兩孔之 n 種匹配狀況示意圖。

其匹配情況有 $1 \rightarrow r, 2 \rightarrow r, \dots, i \rightarrow r, \dots, n \rightarrow r$ 等 r 種狀況發生。

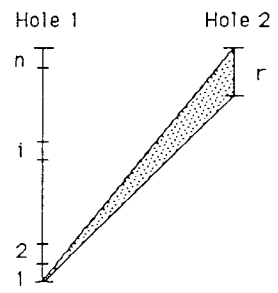
狀況 2：第二鑽孔之土層不與第一鑽孔之任一土層匹配而產生尖滅層，尖點則將出現於 n 個土層之間（圖 4.4a），或 n 層之上端（圖 4.4b），或 n 層之下端（圖 4.4c），因此有 $n+1$ 種可能如圖 4.4。



(a)



(b)



(c)

圖 4.4 首孔有 n 層土層，次孔只有一層土層時，
相鄰兩孔之 $n+1$ 種匹配狀況示意圖。

總計以上兩種狀況，可得 n 層土層與 1 層土層之匹配可能情形為 $2n+1$ 種，即：

$$f(n, 1) = n + (n+1) = 2n+1 \quad (1)$$

(II) 若以第一鑽孔有 n 層土層，第二鑽孔有 m 層土層，討論一般鑽孔情形時，第二鑽孔之第一層土層和第一鑽孔之「任一土層」或「土層間隙」之各種匹配可能性，有如下三種狀況：

狀況 1：首先考慮第二鑽孔之首層無法與第一鑽孔之任一土層匹配，而在第一鑽孔頂端發生尖滅點時，則剩下第一鑽孔之 n 層與第二鑽孔其餘之 $m-1$ 層匹配，其可能之匹配次數可以用遞迴函數 $f(n, m-1)$ 來表示。(見圖4.5)

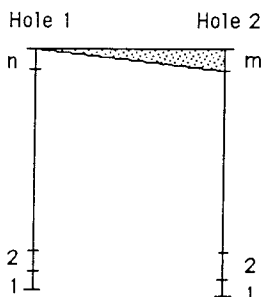


圖 4.5 相鄰兩孔於狀況1之匹配示意圖

狀況 2：當第二鑽孔之首層與第一鑽孔之第 i 層已達成匹配(其中 $1 \leq i \leq n$) 時，則第一鑽孔 i 層以下與第二鑽孔其餘之 $m-1$ 層之匹配可以用遞迴函數 $f(i-1, m-1)$ 表示如圖4.6。

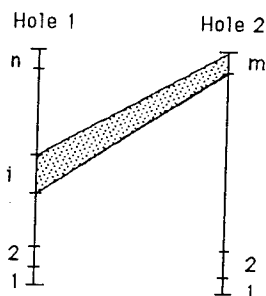


圖 4.6 相鄰兩孔於狀況2之匹配示意圖

狀況3：若第二鑽孔之首層與第一鑽孔之第 i 層下端形成尖滅層(可能情形如第一鑽孔第 i 層以上之土層已與第二鑽孔之第一層上端形成尖滅層時)，其中 $1 \leq i \leq n$ ，則第一鑽孔第 i 層以下土層與第二鑽孔之 $m-1$ 層土層之匹配函數為 $f(i-1, m-1)$ 。(見圖4.7)

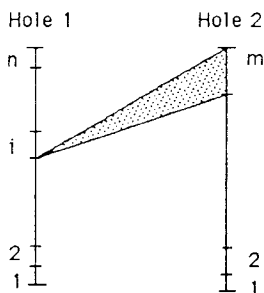


圖 4.7 相鄰兩孔於狀況3之匹配示意圖

綜合上述三項可能發生狀況，可知 n 層土層與 m 層土層匹配之所有可能情形為：

$$f(n, m) = f(n, m-1) + \sum_{i=1}^n f(i-1, m-1) + \sum_{i=1}^n f(i-1, m-1)$$

故

$$f(n, m) = f(n, m-1) + 2 \sum_{i=1}^n f(i-1, m-1) \quad (2)$$

式(1)和式(2)即為構成遞迴函數之兩項基本公式。根據式(1)和式(2)可實際算出 $f(8, 8) = 265729$, $f(9, 9) = 1462563$, $f(10, 10) = 8097453$, $f(11, 11) = 45046719$ 等等，依此類推 $f(12, 12) = 251595969$ ，意即以各含12層土層資料之相鄰兩孔，其土層匹配之種類已高達二億五千萬種以上。

以附錄六所示之鑽孔資料來看 [31]，一個鑽孔往往有數十個採樣資料，而其土壤分類可能有十至三十種之不同，其匹配種類可能達到天文數字之多，故不適合以一般數學窮盡法求取其最佳匹配狀況。這種現象說明了分層之途徑唯有採取經驗法則之推理機構來作分層之分析。

4.2 人工智慧中相關之匹配理論

Aho 與 Peterson 於 1972 年提出字串辨識理論 (String pattern recognition theory) 以解決序列資料之匹配問題。若將每一鑽孔之資料依序排列，計算機便可將其視為有意義之字串，而相鄰兩鑽孔之匹配便可視同兩字串之匹配。在其研究下產生了如下之錯換定義 (error transformation) [1]：

定義一： V^*_t 是字串之集合， $X, Y \in V^*_t$ ，則可定義一種變換 $T: V^*_t \rightarrow V^*_t$ ，使得 $Y \in T(X)$ ，三種基本變換可定義如下：

1. 取代錯換 (Substitution error transformation)

$$w_1 a w_2 \xrightarrow{T_s} w_1 b w_2 ; a, b \in V^*_t, a \neq b$$

2. 遺落錯換 (Deletion error transformation)

$$w_1 a w_2 \xrightarrow{T_d} w_1 w_2 ; a \in V^*_t$$

3. 增誤錯換 (Insertion error transformation)

$$w_1 a w_2 \xrightarrow{T_i} w_1 a w_2 ; a \in V^*_t$$

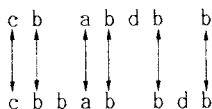
定義二： $X, Y \in V^*_t$ ， X, Y 之距離 $d(X, Y)$ 定義為從 X 變換至 Y 之最小錯換數。

以兩字串 $x=cbabdbb$ 與 $y=cbbabdbb$ 為例，說明如下之匹配方式最佳 [6]：

依照定義一的三種基本變換，字串 X 可利用以下步驟錯換為 Y 字串，而達成上述之最佳匹配方式：

此時，在兩者間之最小差距 (minimal distance) 為 3，其他任何匹配方式之差距均將大於或等於3。例如兩字串直接相匹配時：

前述定義中，「遺落錯換」和「增誤錯換」可視為土層尖滅之形式，而「取代錯換」則是硬將兩種不同土層視為同一層土。因此，除了人為失誤造成資料庫資料不合理之外，在鑽孔資料方面，「取代錯換」不可能發生，故若要引用 Aho 與 Peterson 之匹配方法，則可取消「取代錯換」，且上例之最佳匹配將為：



其錯換之步驟，可能是：

$$\begin{aligned}
 X = cbabdbb & \xrightarrow{T_i} cb\underline{b}abdbb \\
 & \xrightarrow{T_d} cbbabbb \\
 & \xrightarrow{T_i} cbbabdbb = Y
 \end{aligned}$$

此實例之最小差距仍為 3。

於1982強調此種匹配之權重 (Weight) 關係及使用計算機來搜尋最佳匹配之方法，其搜尋理論可藉圖4.8 說明。

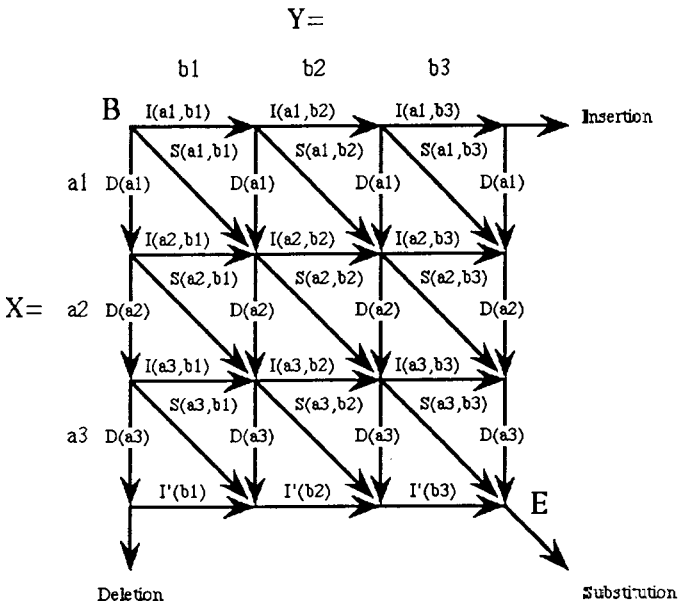


圖 4.8 Fu之權重字串匹配理論之圖解[6]

圖中 $S(a,b)$ 代表以 b 取代 a 所給予之權重。

$D(a)$ 代表取消 a 所給予之權重。

$I(a,b)$ 代表在 a 之前，增列 b 所給予之權重。

$I'(b)$ 代表在 X 字串之後，增列 b 所給予之權重。

此處之權重可視為是兩字串差異性之表徵。

圖上每一個交點顯示出各種搜尋狀態。左上角之“B”表示開始匹配之狀態，右下角“E”表示完成匹配的狀態。從B點出發經由各種路徑到達E點，即達成一次匹配。在人工智慧裡希望模擬出一種合理的方式，意即在合理的時間內，找尋出一種可接受的匹配。故本研究乃以“找出一種合理的，且可令人接受的相鄰鑽孔土層資料之劃分方式”為目的。

圖4.8顯示兩字串 $X=\{a_1, a_2, a_3\}$ 與 $Y=\{b_1, b_2, b_3\}$ 的各種匹配路徑，從B點開始運作，B點右側橫軸上有 b_1, b_2, b_3 是Y字串之組成，B點下方縱軸有 a_1, a_2, a_3 是X字串之組成。其推理的程序如下所示：

開始時兩字串之差異值假設為零，並觀察此狀態下兩字串之首字元 a_1 和 b_1 。其可能狀況有二：

- 一. 若 a_1 和 b_1 相同，則差異值不變，B點之狀態轉變為B點下斜線段之另一端點狀態，並且開始新狀態下之兩字串 $\{a_2, a_3\}$ ， $\{b_2, b_3\}$ 之比較。
- 二. 若 a_1 和 b_1 不同，則有三種情況可考慮。

情況1：第一字串之 a_1 之前應補入 b_1 (意即「增誤錯換」之定義)，此時，差異值提高了 $I(a_1, b_1)$ 之值，X, Y字串之 b_1 達成匹配，B點之狀態轉變為B點右側橫線段之另一端點狀態，然後開始新狀態下之兩字串 $\{a_1, a_2, a_3\}$ 與 $\{b_2, b_3\}$ 之比較。

情況2：第一字串之 a_1 被認定是誤增之雜訊而需移除(意即「遺落錯換」之定義)，此時差異值提高了 $D(a_1)$ 之值，B點之狀態轉變為B點下方直線段之另一端點狀態，然後開始新狀態下之兩字串 $\{a_2, a_3\}$ 與 $\{b_1, b_2, b_3\}$ 之比較。

情況3：第一字串之 a_1 與第二字串之 b_1 ，被認定是兩者錯換造成的(意即「取代錯換」之定義)，而將X字串的 a_1 改變成此時差異值提高了 $S(a_1, b_1)$ 之值，X, Y字串之 b_1 達成匹配，而進入B一端點狀態，然後開始新狀態下兩字串 $\{a_2, a_3\}$ 與 $\{b_2, b_3\}$ 之之比較。

如此從一狀態進入另一狀態之過程一直進行下去，直到進入E端點之狀態，才達成一種匹配比較的過程。透過各種匹配路徑，可以比較其個別之差異值，選出最小者，作為最佳匹配方式

本研究修正Aho 與Peterson之錯換定義取消「取代錯換」存在之可能性，作為在兩鑽孔間比較土層資料並決定土層劃分之方法，故圖 4.8 可修正如圖4.9，作為描述兩鑽孔土層資料 $X=\{a1, a2, a3\}$ 與土層資料 $Y=\{ b1, b2, b3\}$ 之各種匹配路徑。其推理程序相同於本節之前段所述，但是必須刪除「取代錯換」之情況，並且當兩土層資料 $a1$ 與 $b1$ 比較時，要考慮適當的彈性。

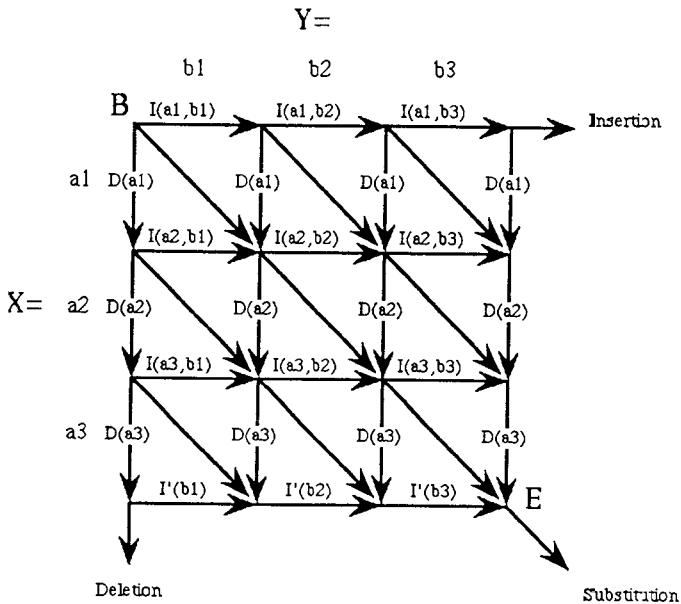


圖 4.9 本研究之兩鑽孔土層資料之匹配程序圖解

對於任一字串而言，其內含之每一字元都非常明確，但是，對於任一鑽孔內之每一土層中尚可能含有數個土壤細分層。因此，在土壤層次之資料匹配方面，其複雜程度遠超過 Aho和 Peterson，以及Fu所考慮之可能性。因此，本文除了修改前人所提之錯換理論，另行發展一套新的模型以外，並依第 4.4節將說明之「層次相似值」定義等建立一套推理機，此模型與推理機將分述於後。

4.3 鑽探基本資料之表達

鑽探基本資料為本專家系統之分析對象，其必須有適當的表達方式，以利推論機之運作。故在專家系統存有一資料擷取轉換介面，將資料取出並轉換成所需適當之表達方式。其以「形式語言」(Formal language) 表達之方式如下：

$$\begin{aligned} \text{土岩區分} = & \text{“土岩分類”} + \text{“N 值”} + \text{“礫石百分比”} \\ & + \text{“砂土百分比”} + \text{“粉土百分比”} \\ & + \text{“黏土百分比”} + \text{“液性限度”} \\ & + \text{“塑性指數”} + \text{“起深”} + \text{“止深”}。 \end{aligned}$$
$$\text{層次區分} = \{ \text{說明} \} + \{ \text{土岩區分} \}。$$
$$\begin{aligned} \text{鑽孔} = & \text{“X 座標”} + \text{“Y 座標”} + \text{“Z 座標”} \\ & + \{ \text{層次區分} \}。 \end{aligned}$$

其中「+」表示資料之連續性，「“ ”」表示內含一項不重複資料之單元，而「{ }」表示其內資料有多次重複之可能性。例如{層次區分}表示一鑽孔內之層次有可能有幾層。

對於錐形發展方式而言，必須在簡單明確之架構下，以反覆實驗設計和逐漸擴充的程序建立系統。故在發展原形(prototype)初期的推理機裡，僅考慮鑽孔的層次深度，土岩分類，N 值之簡單關係。

以Turbo Prolog之語法所描述出的資料結構可表示如下：

```
Hole      = Layer*
Layer     = segment*
Segment  = segment(string,integer,real,real)
          % segment(Class,N_Value,BeginDepth,EndDepth)
```

4.4 土岩對等值與層次相似值之介紹

土／岩層之匹配，基本上若兩層之土壤相同，即可予以匹配。但土壤不同則未必就不能匹配，有些土壤雖然組成不同，其性質卻相似，因此在專家的眼裡可能將此不同組成的土壤予以匹配而成為同一土層。如黏土的CL與ML雖不同類，但在某方面可視為相似。

因此本研究提出對等值與相似值之概念，以做為評估土層相似性及鄰孔間相似性之依據。

本研究中孔層之土壤內容是以重元集(Multiset)的概念表示，因此以下即先描述重元集。

一、重元集

數學上的集合被定義成「含有明確且相異元素的群體」，此和土層是一群土岩的集合，並不太相同。若假設某一土層之組成份子為"CL"，"CL"，"ML"，"CL"，"CL"，若將此土層視為集合，而土壤類別為元素，則此集合即為{"CL"，"ML"}，此集合雖可表示此土層之組成內容，但卻無法表示出土層內組成份子之序列關係(層次關係)。

為了表示土層內組成份子之序列關係，乃引用離散數學(Discrete mathematics)之重元集(Multiset)理論表示之。所謂重元集(Multiset)是由一群不一定相異之事物組成[13]。例如 {a,a,a,a}，{a,a,a,b,c}，{a,a,b,b}，{} 和 {a,b} 都是重元集，因此，前段所述集合的定義可視為重元集的特例。一個重元集中，某一元素之重次(Multiplicity)係指此元素在此重元集中重複出現的次數。例如{a,a,c,d,d,d}中，a的重次為2，b的重次為0，c的重次為1，d的重次為3。

前述所提之土層組成集合若改以{"CL", "CL", "ML", "CL", "CL"}或{"CL", "ML", "CL"}之重元集表示方式，則既可表示其組成也可表現出其序列關係，即上層為"CL"，中層為"ML"，而底層為"CL"。

二、對等值

土壤性質之獲得是由實地鑽探試驗而得，因此相同的土壤可能會因天候，因人之不同而得不同之試驗結果。而原來同一土層之土壤也可能因區域，因環境之影響而性質會稍有不同，但卻應被視為同一土層。因此土層匹配應以土壤相同為優先考慮，而土壤相似者次要考慮之。

對等值之制定，即由兩土壤間性質之相似性予以不同等級之權重，原則上以性質愈相似者權重(即對等值)愈高，權值之給定即須土質專家之協助，依其在土壤學上專業的知識，給予兩兩不同之土壤應有之對等值。詳見附錄一。

相同之土壤亦會因強度之不同，而被分成不同的土層，本研究於強度之參考數據暫以N值(N Value)為依據，如表4.1 與4.2：[32]

表4.1 黏性土之強度數據

稠 度	N 值範圍
極軟弱	< 2
軟 弱	2 - 4
中等堅實	4 - 8
堅 實	8 - 15
極堅實	15 - 30
堅 硬	> 30

表 4.2 無黏性土之強度數據

硬 度	N 值範圍
極疏鬆	0 - 4
疏 鬆	4 - 10
中等緊密	10 - 30
緊 密	30 - 50
極緊密	< 50

由上表可知，N 值差異愈大，代表土壤之強度差異愈大，因此兩相同土壤間會因N 值之差異而影響兩土壤原來之性質，因此本研究除了對等值之考慮之外，也將考慮土壤強度對土層匹配之影響。詳見附錄二。

以下即是利用Turbo Prolog語法表示出上述兩種知識之事實(Fact)範例：

```
Eq_sub11("GW", "GW-GM", 95)
Eq_sub12("GW", "SW", 30)
Eq_sub13("GP-GM-GC", "ML", 1)
Eq_sub22("SW", "SW-SM", 90)
Eq_sub23("SW", "ML", 10)
Eq_sub33("ML", "CL-ML", 90)
Eq_sub_N1(2, 4, 85)
Eq_sub_N2(0, 4, 85)
```

本研究中對等值之分配是由1到100之整數，以上各事實之第三個欄位即此值。附錄三則附有詳細之知識表示。

三、層次相似值之計算模式

土壤分層是以層次 (Layer) 為基本單位，而層次則是由一些土／岩組成，鑽孔是由一些有秩序的層次組合，其組合以形式語言表示如下：

hole : Δ layers (多層土層組成鑽孔)

layer : segments (多層土壤段落形成單一土層)

各孔之原始分層是以大地工程地質資料庫的鑽孔資料為準，層次的匹配與否是以層次之間相似值之大小為依據，以下即為本研究之層次相似值計算模式。

首先將計算模式之符號解釋如下：

seg(i,j,k) : 第i鑽孔，第j層次之第k個土壤段落。

layer(i,j) : 第i鑽孔，第j段之土層。

Equality(seg(i,j,k) : seg(u,v,w)) : 兩土壤段落之對等值，以整數表示。

Similarity(seg(i,j,k) : layer(u,v)) : 某土壤段落與某土層之相似值，以整數表示。

Similarity(layer(i,j) : layer(u,v)) : 兩土層之相似值，以整數表示。

i , u : 鑽孔編號

j , v : 土層編號

k , w : 土壤段落編號

(a) 某土壤段落與某土層之相似值

$$\text{Similarity}(\text{seg}(i, j, k) : \text{layer}(u, v)) =$$

$$\frac{\sum_{w=1}^{u,v} \text{Equality}(\text{seg}(i, j, k) : \text{seg}(u, v, w))}{N_{u,v}}$$

如某一土壤段落"CL"與某土層 {"CL", "OL"}
之相似值為

$$\begin{aligned} & \text{Similarity} ("CL":\{"CL", "OL"\}) \\ &= \frac{\text{Equality} ("CL":"CL") + \text{Equality} ("CL":"OL")}{2} \\ &= \frac{100 + 80}{2} = 90 \end{aligned}$$

(c) 某土層與某土層之相似值

本研究之層次相似值的求得有二個途徑：

- 一、若兩土層內土壤段落之內容與順序完全相同者，令其相似值為100。
- 二、否則即以下述之層次相似值計算模式計算之。

$$\text{Similarity}(\text{layer}(i, j) : \text{layer}(u, v)) =$$

$$\frac{\sum_{k=1}^{N_{i,j}} \sum_{w=1}^{N_{u,v}} \text{Equality}(\text{seg}(i, j, k) : \text{seg}(u, v, w))}{N_{i,j} * N_{u,v}}$$

$N_{i,j}$ ：第*i*鑽孔，第*j*土層之土壤段落個數

$N_{u,v}$ ：第u鑽孔，第v土層之土壤段落個數

以二土層 {"CL", "OL", "CL"} 與 {"CL", "OL", "CL"} 而言，根據途徑一，可得其相似值為100。

若二土層為 {"CL", "OL"}， {"CL", "CH"} 則相值為：

Similarity($\{-CL\}, \{OL\}; \{-CL\}, \{CH\}$)

$$= \frac{[Equality(\{-CL\}, \{CL\}) + Equality(\{-CL\}, \{CH\})] + [Equality(\{OL\}, \{CL\}) + Equality(\{OL\}, \{CH\})]}{2 * 2}$$

$$= \frac{[100 + 50] + [80 + 60]}{4} = 72.5$$

本研究之原形系統，即依此二法則計算相似值，以 Turbo Prolog 語言描述之計算法則如下：

```

Similarto(Layer1,Layer2,Simi,0):-                               途徑一
    Similar_string(Layer1,Layer2),
    largest_weight(Simi),!.
Similarto(Layer1,Layer2,Simi,Unsimi):-                         途徑二
    Accumulate_Weight_of_Intersect(Layer1,Layer2,Accum_Simi),
    Count_segment_number(Layer1,N1),
    Simi = round(Accum_Simi / N1),
    Largest_weight(Largest_weight),
    Unsimi= Largest_weight - Simi.

Accumulate_Weight_of_Intersect([],_,0):-!.
Accumulate_Weight_of_Intersect([X|L1],L,Weight):-
    MemberWeight(X,L,Accum_Weight1),
    Count_segment_number(L,L_No),
    Weight1 = Accum_Weight1 Div L_No,!.
    Accumulate_Weight_of_Intersect(L1,L,Weight2),
    Weight = Weight1+Weight2.

MemberWeight(_,[],0):-!.
MemberWeight(X,[Y|Tail],Weight):-
    Eq(X,Y,Weight1),!,
    MemberWeight(X,Tail,Weight2),
    Weight=Weight1+Weight2,!.

Count_segment_number([],0):-!.
Count_segment_number(_|L1],C):-
    Count_segment_number(L1,CC),
    C = CC + 1.

```

在計算相似值之同時亦計算兩鑽孔之相異值以作為往後兩鑽孔相似值比較之用。

4.5 推理機之規則

本研究係模擬人類專家對土壤分層之推理行為，而以 Turbo Prolog 語言建構了一外建之推理機，並以 Turbo Prolog 本身的內建推理機驅動此一外建推理機。

相鄰兩鑽孔之土層匹配推理法則概述如下：

- 1、算出兩孔之相似值。
- 2、以某孔為待分割孔，另一孔為參考孔，測試待分割孔是否有必要再分割，使分割後之鑽孔與參考孔重新比較後，得較高之相似值。
- 3、將待分割孔與參考孔之角色調換，再行比較分割。

以下即針對整個推理機之各推理過程與考慮，作一番描述。

4.5.1 最短路徑法(Mini Path)

本研究在鑽孔間的土層匹配，可以最短路徑法的觀念，描述其匹配的原理。

為明瞭兩鑽孔間之土層匹配路徑，吾人可將土層匹配時有可能之路徑應用於圖形理論(Graph Theory)表示，以各為兩層之鑽孔為例，如圖4.10。

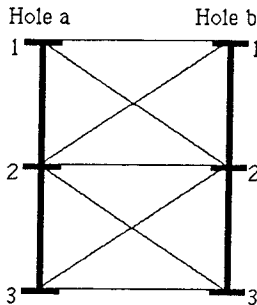


圖 4.10 各有兩層次之相鄰二鑽孔，
土層匹配之所有狀況。

其在匹配前之起始狀態係將兩鑽孔頂層之上端點先行連接，即 Hole a 編號為 1 之點與 Hole b 編號為 1 之點連接，以 "a1-b1" 表示。而搜尋之最後狀態則是將兩鑽孔底層之下端點連接即 Hole a 編號為 3 之點與 Hole b 編號為 3 之點連接，以 "a3-b3" 表示。其間之所有可能路徑則以圖 4.11 表示。

圖 4.11 之節點 (Vertices) 部份，表示層次匹配分層時，分層之界線可能連接之情形。有箭頭之邊線 (Edge)，則代表匹配之程序，兩土層匹配時所產生之相似值與相異值即表示於邊線上，代表欲經過此邊線所給予之權重。由起始狀態經過一些節點與邊線而至最後狀態即形成搜尋之路徑 (Path)。不同之路徑有不同之權重，而其權重係由此路徑所經過之邊線所代表之相似值與相異值累加而成。欲獲得較好之土層匹配，即須選擇相似值較高而相異值較低之路徑，因為路徑之構成，即代表兩鑽孔土層之組合。

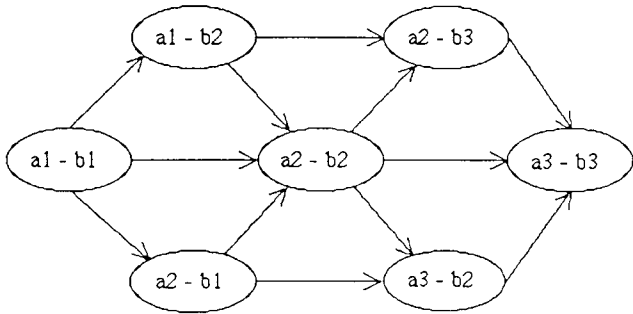


圖 4.11 兩鑽孔匹配之圖形表示

本研究即是以這些路徑為基礎，以一經驗法則找出相似值較高而相異值較低之路徑。

以一鑽孔為"CL", "CH"兩土層與另一鑽孔土層有"OL", "MH"為例(如圖4.12), 其二鑽孔匹配之圖形表示為圖4.13所示。由於各土層僅有一種土壤, 因此各土層相似值恰等於各土壤之對等值(參見附錄一)。綜合各路徑之權值可得權重為180之路徑為較適當之分割路徑。

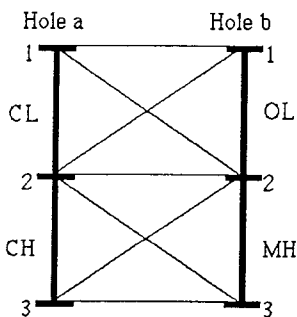


圖 4.12 相鄰二鑽孔之各種分割情形

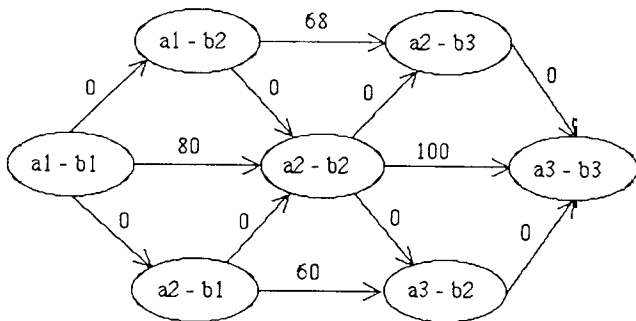


圖 4.13 兩鑽孔匹配之權重與圖形表示

路徑	路徑中之節點	各路徑之權重
1	a1-b1, a1-b2, a2-b3, a3-b3	68
2	a1-b1, a1-b2, a2-b2, a2-b3, a3-b3	0
3	a1-b1, a1-b2, a2-b2, a3-b3	100
4	a1-b1, a1-b2, a2-b2, a3-b2, a3-b3	0
5	a1-b1, a2-b2, a2-b3, a3-b3	80
6	a1-b1, a2-b2, a3-b3	180
7	a1-b1, a2-b2, a3-b2, a3-b3	80
8	a1-b2, a2-b1, a2-b2, a2-b3, a3-b3	0
9	a1-b1, a2-b1, a2-b2, a3-b3	100
10	a1-b1, a2-b1, a2-b2, a3-b2, a3-b3	0
11	a1-b1, a2-b1, a3-b2, a3-b3	60

4.5.2 鄰孔比對

鄰孔比對即以兩土層相似值為基礎，進而推算兩鑽孔之相似值。

兩鑽孔之相似與否，在於其鑽孔間匹配的土層之多寡。若兩鑽孔之土層均能依序匹配，則其相似性應屬較高，而相似值當然也較高；若有較多之土層無法匹配，而發生較多的尖滅層，則代表此二鑽孔相似性較低。且上層土層若與鄰孔匹配成功後，即不能再與下層繼續匹配。

如圖4.14，若Hole1之第一層與Hole2之第二層匹配成功後，應繼續推斷Hole1 第二層以下與Hole2 第三層以下之土層。

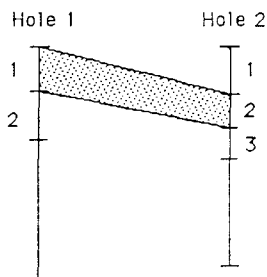


圖 4.14 兩鑽孔之土層匹配

兩鑽孔之相似值即針對所有匹配成功之土層，將兩相匹配之土層的相似值累加起來，土層相似值即由 Aho 與 Peterson 之字串辨識理論中各種錯換所得的權值衍生而來。兩鑽孔之相似值推理法則可敘述如下：

- 1、若Hole1 之土層均匹配完畢，則計算Hole2 所餘尚未匹配土層之數目，視數目多寡增加兩鑽孔之相異值，並結束推理。如圖4.15。

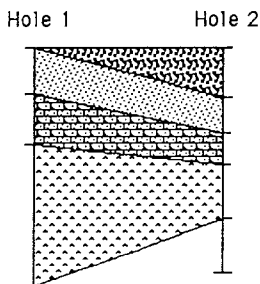


圖4.15 兩鑽孔匹配，首孔之土層已全部匹配完成。

- 2、若Hole1尚餘一些土層未匹配，而Hole2之土層卻已匹配完畢，則依Hole1所餘土層之多寡，增加兩鑽孔之相異值，並結束推理。如圖4.16。

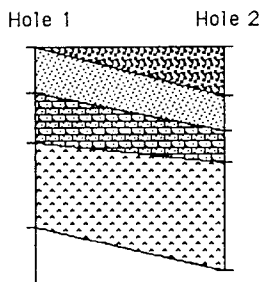


圖4.16 兩鑽孔匹配，次孔之土層已全部匹配完成。

- 3、若兩鑽孔中尚未匹配之土層的頂層，相似值夠高，可匹配為同一土層，則計算此兩頂層之相似值與相異值，並移去此二頂層，繼續遞迴推理所餘之土層。如圖4.17

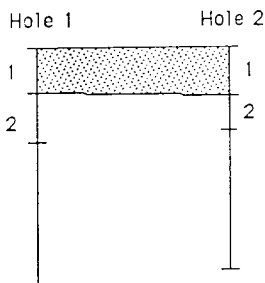


圖4.17 兩鑽孔之頂層完成匹配

- 4、若前三項皆不滿足現行狀況，則分兩狀況推理：
- (1) 只移去Hole2 之頂層，繼續遞迴推理所餘之土層，並計算相似值與相異值。如圖4.18a。
 - (2) 只移去 Hole1之頂層，繼續遞迴推理所餘之土層，並計算相似值與相異值。如圖4.18b。然後取較大之相似值累加至兩鑽孔之相似值；將相對應之相異值，累加至兩鑽孔之相異值，並結束推理。

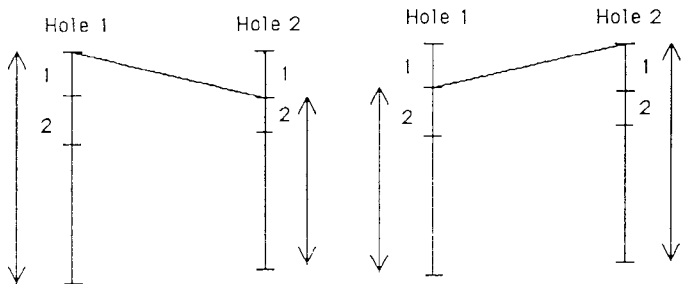


圖 4.18a 移去Hole2之頂層

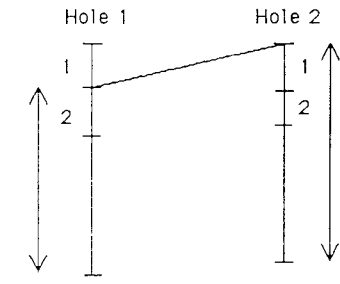


圖 4.18b 移去Hole1之頂層

以上法則，以Turbo Prolog 描述，即為以下程式之 evaluate_difference_and_same_value_of_hole_between_hole 之宣敘式。

```

evaluate_difference_and_samevalue_of_hole_between_hole([],Hole2,Dist,0) :-      法則1
    overlength_of_hole(Hole2,Dist),!.
evaluate_difference_and_samevalue_of_hole(Hole1,[],Dist,0) :-      法則2
    overlength_of_hole(Hole1,Dist),!.
evaluate_difference_and_samevalue_of_hole([X|Hole1],[Y|Hole2],Dist,Same) :-
    SimilarHorizon(X,Y,Simi,Unsimi),
    ref_weight(Ref_weight),
    SimilarCriteria(Simi,Ref_weight),!,
    evaluate_difference_and_samevalue_of_hole(Hole1,Hole2,Dist1,Same1),
    Dist = Dist1 + Unsimi,
    Same = Same1 + Simi,!.
evaluate_difference_and_samevalue_of_hole([X|Hole1],[Y|Hole2],Dist,Same) :-
    SimilarHorizon(X,Y,Simi,Unsimi),
    ref_weight(Ref_weight),
    SimilarCriteria(Simi,Ref_weight),!,
    evaluate_difference_and_samevalue_of_hole(Hole1,Hole2,Dist1,Same1),!,
    evaluate_difference_and_samevalue_of_hole(Hole1,[Y|Hole2],Dist2,Same2),!,
    dcid_Dist_Same(Dist1,Dist2,Dist,Same1,Same2,Same). % meta knowledge
    法則4

```

4.5.3 層次細分

從大地工程地質資料庫取出之鑽孔資料，其層次區分只是基於同一孔洞資料之考慮。若進一步與鄰孔之資料比較，該孔之某些層次區分可能必須調整與細分，才能使孔層間的匹配有較好之結果，即土層之剖面會有較合理的分層。

推理機在執行推理時之步驟為

1. 對各個層次考慮是否值得細分。
2. 若值得細分，則進一步檢驗細分後和鄰孔比對之結果。
3. 若有較佳之比對結果（即有較高之鑽孔相似值），則接受層次細分，否則原層次不變。

層次細分之推理，係以大中取小(MINIMAX)之控制策略改寫而得，上述之步驟1 即找出各層次內之各段落之間相異值最大者，作為層次細分之點；步驟3 所謂較佳之比對結果，即代表找出相異值較小者。

判斷層次是否值得細分之推理，即上述推理步驟之第一步，可如下表示：

```
BestCut(Segment1,Segment2,Reference_Weight):-  
  SimilarVertical(Segment1,Segment2,Simi,_),  
  CutCriteria(Simi,Reference_Weight),  
  cutedRec(_,_,OldSimi),  
  Simi < OldSimi,  
  input_cutedRec(Segment1,Segment2,Simi).
```

這種判斷是否值得細分之法則，可另行加入其他之判斷法則，以適合特殊的需要。

如步驟2 所述，當一層次細分後，與鄰孔比對而使兩鑽孔相似值之增幅大於其相異值之增幅，代表該層次必須細分，而將調整該孔之分層關係。這種較好之關係之決定，使用了人工智慧之「配對經驗搜尋法則」(Matching rule)，其部份推理如下：

```
fitHole(RefHole,OldHole,NewHole,OldDist,Same):-  
  repeat_fit_segments(RefHole,OldHole),  
  cuttingRec(NewHole,OldDist,Same).  
  
repeat_fit_segments(RefHole,OldHole):-  
  append_layers(Up,[Mid_layer|Bottom],OldHole),  
  pack(Mid_layer,Packed_layer),  
  not(check_if_pure_layer(Packed_layer)),  
  fitsegment(RefHole,Up,Mid_layer,Bottom),  
  fail.  
repeat_fit_segments(_,_).
```

4.5.4 多孔層次之匹配

通常土層剖面分析時，所給予的鑽孔資料不只兩孔，因此鑽孔資料分層時的細分可能發生漣漪效應，即一孔之資料可能不只影響到鄰孔資料之分層，也可能透過影響鄰孔之現象，繼續影響到更遠之孔的分層。因此推理機構應容許循迴檢驗交考慮多孔間分層匹配的問題。本推理機是透過如下的程序來達成多孔層次的匹配問題：

```
matching_hole_from_left:-
```

```
temp_lr_layers_content_of_hole(No,CordX,CordY,Begin,End,CuttingHole),
next_right_hole(No,RefHole),
evaluate_difference_and_samevalue_of_hole(
                                CuttingHole,RefHole,Dist,Same),
input_cuttingRec(CuttingHole,Dist,Same),
try_to_make_a_bestfitting(RefHole,CutedHole),
asserta(temp_lr_layers_content_of_hole(
                                No,CordX,CordY,Begin,End,CutedHole)),
fail.
```

```
matching_hole_from_left:-!.
```

```
matching_hole_from_right:-
```

```
temp_rl_layers_content_of_hole(No,CordX,CordY,Begin,End,CuttingHole),
next_left_hole(No,RefHole),
evaluate_difference_and_samevalue_of_hole(
                                CuttingHole,RefHole,Dist,Same),
input_cuttingRec(CuttingHole,Dist,Same),
try_to_make_a_bestfitting(RefHole,CutedHole),
asserta(temp_rl_layers_content_of_hole(
                                No,CordX,CordY,Begin,End,CutedHole)),
fail.
```

```
matching_hole_from_right:-!.
```


Matching_Layer_from_left 之功能為由最左邊鑽孔向右邊鑽孔漣漪擴散，而Matching_Layer_from_right之功能則由最右邊鑽孔向左邊方向漣漪擴散，周而復始，多次匹配後達到多孔層次之最佳匹配。

4.5.5 土層劃分之決定

經過多孔層次之匹配後，可得到各孔在較佳匹配下之分層。如此便可將鑽孔間之土層匹配表示出來，以一既定的資料結構傳給土層剖面圖顯示界面，將推論之分層結果，以圖形的方式表示出來。其推理邏輯，則與上述的鄰孔比對相同。

在此程序中，尚須考慮N 值對於分層之影響，亦即在土層匹配時，考慮同一層次中因N 值之變化大小而決定是否再細分層次。

附錄四表示了本研究整個推理機之核心部份。

五、土壤分層專家系統之功能

本系統之主要功能係以專家之經驗法則推論出土層之剖面圖，其以專家系統為架構，保留一般專家系統所擁有的使用者界面、知識庫、推理機、工作記憶區，並增加外部資料擷取界面，而成為本系統之架構。本系統推論所依據之鑽孔資料係源自大地工程地質資料庫之部份地質資料；所依據的知識則包含土層剖面圖分層之推論法則及推論所依據之土壤對等值、 N 值對等值。

5.1 工具選擇

本系統是在MS-DOS的作業環境下，以Turbo Prolog程式語言與其提供的 Toolbox 建構而成。

執行此專家系統所必須的配備有：

- IBM PC或其相容性電腦。
- PC-DOS或MS-DOS 2.0 版本以後之作業系統。

Turbo Prolog 是一種宣告式(declarative)語言，也就是說，給它必要的事實與規則，它就可用推理的方法來解決程式問題。相反地，如Pascal,Basic和其它傳統的電腦語言是程序化(Procedural)的語言：程式設計師必須提供一步一步的程序來告訴電腦如何去解決問題。TurboProlog的程式設計師只須提供問題(或目標 goal)的描述，與解決問題的基本規則，Turbo Prolog系統便會決定如何去找答案。[21]

Turbo Prolog 的一些重要性質有：

1. 由Turbo Prolog所引用之事實或規則，完全是由程式設計師所給定，它無從判斷這些知識(即事實與規則)的完整性與可靠性，因此Turbo Prolog只好假設其推理的問題領域內所獲得的知識是正確且完整的。

- 2.當Turbo Prolog程式在運作時，其函數功能被引用與否，原則上端看目標條件是否滿足，而非由其他函數任意呼叫。
- 3.Turbo Prolog是一目標導向語言，Turbo Prolog程式，只要描述目標，則系統就會自動選擇既有的宣告或事實，以設法找尋到滿足目標的宣告或事實。
- 4.Turbo Prolog系統使用先深度搜尋及後向鏈結作為其主要的推論控制策略。

本研究利用Turbo Prolog之主要原因，係由於它強大的邏輯處理能力，及提供許多方便的界面工具程式，使得系統的開發能較為方便且好用。

5.2 整體架構

為配合大地工程地質資料庫之使用，本系統將分為五個主要組成：使用者界面、推理機、工作記憶區、知識庫、知識編輯模組及外部資料轉換模組。各模組之關係如圖5.1 所示。

以下即概述各模組之內容。

5.2.1 使用者界面 (User Interface)

本系統之使用者界面包括：

- 1、螢幕驅動(Menu Driven)
- 2、圖形顯示界面
- 3、理由解釋界面

本界面係做為使用者與系統之交談媒介，其是以螢幕選項的方式，以驅動各子功能執行其任務。

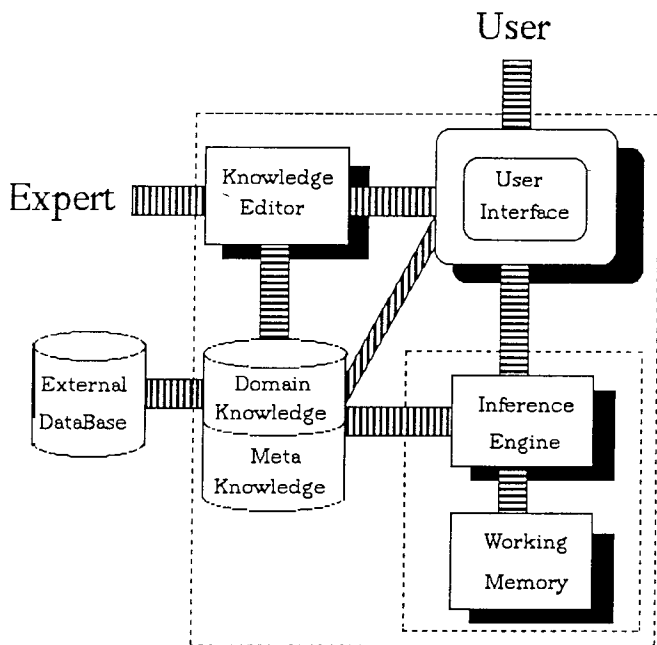


圖 5.1 土層剖面圖專家系統之架構

土層剖面圖之分析結果，是以圖示的方式顯示出土層剖面圖，讓使用者容易地了解系統推論後所得之結果。圖形顯示界面所表示之內容包括：欲推理之各鑽孔之順序、各土層之深度、土壤類別以不同圖樣 (Pattern) 表示與土層之配對情形。

本研究之推理係完全以相似值之大小為基礎，以決定相鄰兩孔之土層是否匹配。因此本系統將推理過程中兩兩土層配對時所依據之相似值於土層剖面圖顯示的同時，亦列於螢幕上，以解釋土層相連的原因。

5.2.2 推理機 (Inference Engine)

本系統所用以建構之人工智慧語言係為 Turbo Prolog。Turbo Prolog 語言擁有內建之推理機，並且採取先深度搜尋 (First Depth Search) 之策略從事知識搜尋之工作；其內建之推理則採後向鏈結 (Backward Chaining) 之推論方式，但也可給予產出規則 (Production Rule)，使其進行前向鏈結 (Forward Chaining) 之推論方式。

因此本系統之推理機，係以 Turbo Prolog 之內建推理機為工具，撰寫產出規則，建構一適合土壤分層推理之外建推理機。第四章所述的「鄰孔比對」、「層次細分」、「多孔層次之匹配」等等，均屬本系統之外建推理機部份。

5.2.3 工作記憶區 (Working Memory)

專家系統運作時，常會推導出一些暫時性的事實，而這些事實有利於系統繼續之推導，其中包括 Turbo Prolog 之內建推理機在推導過程中所產生的事實，及土壤分層之外建推理機所產生的事實。如多孔層次匹配時，儲存各孔分割情形的資料庫。或層次細分時，在未決定細分與否時之暫時細分情形等等。可由下述的資料庫描述表示之：

```
global database-working_memory
cuttingRec(Layers, integer, integer)
cutedRec(Layer, Layer, integer)
temp_rl_layers_content_of_hole(integer, string, string, real, real, Layers)
temp_lr_layers_content_of_hole(integer, string, string, real, real, Layers)
layerpairs_between_two_holes(integer, Layer_pairs)
```

5.2.4 知識庫 (Knowledge Base)

本系統之知識庫可概分為領域知識庫 (Domain knowledge Base) 與層上知識庫 (Meta knowledge Base)。

第四章所述相似值之計算完全根基於土壤對等值，不同配對之土壤會產生不同之對等值，這些對等值即由專家給定而放置於領域知識庫。

相似值計算後要如何認定此值代表相似或相異呢？當然須有一準則參考值藉以判定相似與否，若相似值大於此參考值則視為相似，因此領域專家須依其給定對等值之原則，定出一準則參考值於領域知識庫內。

層上知識庫即存放影響推理路徑之決策法則。如判斷土層是細分或相似之準則：

若相似值小於比較參考值，則細分之。

若相似值大 於或等於準則參考值，則 視為同一土層。

以Turbo Prolog語法表示則為：

```
CutCriteria(Simi,Reference_Weight):-  
    Simi < Reference_Weight.
```

```
SimilarCriteria(Simi,Reference_Weight):-  
    Simi >= Reference_Weight.
```

5.2.5 知識編輯模組 (Knowledge Editor)

系統目前係以編輯的方式，將專家在特定領域的知識，依適當的表達方式存入知識庫中。

在知識工程師和專家對談及重複使用專家系統的過程中，常會發現原有的舊知識有修正的必要，因此需要知識編輯模組的協助，以維護知識的正確性，以避免發生知識重複與矛盾之現象。

5.2.6 外部資料轉換模組 (External Data Conversion Model)

本研究係根據大地工程地質資料庫之部份鑽孔內容與特殊之推理機構，記以專家所提供之經驗法則與事實，推導出土層剖面圖。因此為獲取大地工程地質資料庫之相關資料，必須利用一轉換模組從資料庫萃取出有用之資料，以利系統之運作。

本系統之外部資料轉換模組，即將大地工程地質資料庫之土壤分層相關資料轉換成專家系統所能使用之資料型態。且本系統只須用到資料庫中檔案 TABLEA.DBF 內之 ID、H_DEPTH、DESCRIPT、CLASSIFICA、N_VALUE等資料項目，因此即參照此檔案之格式而利用以下之副程式擷取上述之資料項目。

```
rd_Dbase3Rec_from_holeaFile(  
    [ID,H_DEPTH,DESCRIPT,CLASSIFICA,N_Value]):-  
    read_strArr(13,ID),  
    read_strArr(6,H_DEPTH),  
    read_strArr(9,DESCRIPT),  
    read_strArr(5,TempCLASSIFICA),  
    read_strArr(3,N_Value).
```

5.3 系統功能與實作

本系統包含之功能有：1.由選定的鑽孔推斷其間土壤之分層情形；2.判斷兩層次異同之程度；3.判斷土層是否須再細分；4.判斷一鑽孔相對於參考鑽孔是否可再細分；5.外部資料擷取模組；6.知識編輯模組。因此交談界面之建構即採用螢幕選取 (Menu Driven)的方式，以方便使用者之使用。其螢幕設計格式則如圖5.2 所示：

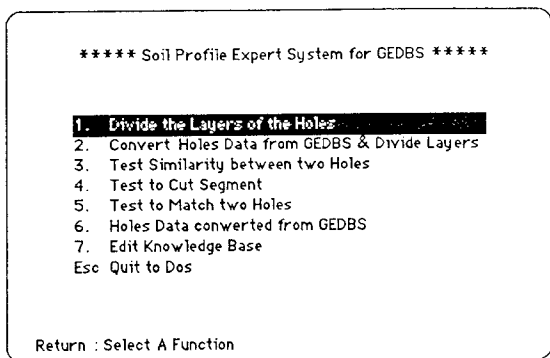


圖 5.2 土層剖面圖專家系統之主功能表

茲將此六項功能分述如下：

(1) 土層分析子系統

於圖5.2之畫面時，當使用者將反白游標移到“Divide The Layers of The Holes”按Return鍵或者按數字鍵“1”，則表示選擇第一項功能，此時螢幕則顯示另一選擇畫面如5.3 圖。

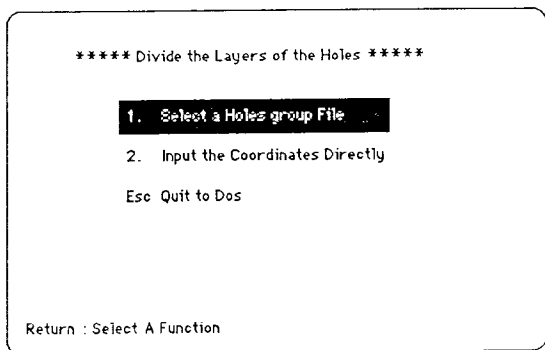


圖 5.3 鑽孔分層功能表

本系統為方便使用者，第一種是利用已建立的檔案（其附加檔名為.HID）是當使用者已知鑽孔座標值。

因此當使用者在圖 5.4 即呈現如下情形，以請

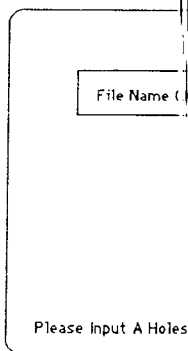


圖 5.4

此時使用者可直接輸入附加檔名（即“.HID”）時，可在冒號後直接按的鑽孔座標檔，而產生

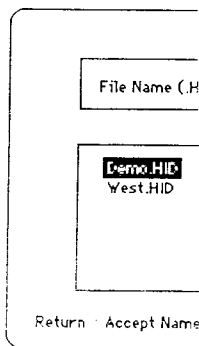
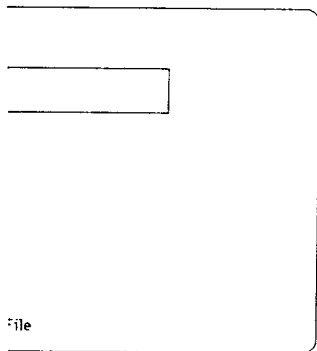


圖 5.5

使用者即可利用反白游

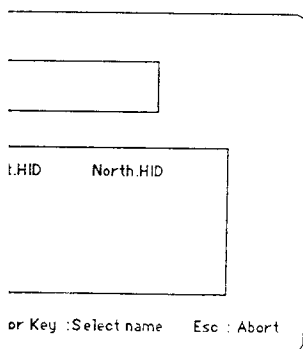
操作，特提供了兩種選取鑽孔的工程地質資料庫系統設計好的座標以作為分層時選孔的依據；另一種時，則可由本系統直接輸入鑽孔座

之畫面，選擇第一項功能時，畫面用者欲輸入之座標檔名。



空標檔案輸入畫面

其欲選擇之檔案名稱，唯不須再鍵使用者不確定其欲選擇之檔案名稱按n鍵，系統即立刻找尋磁碟內所有之畫面。



空標檔案選擇畫面

其所需要的鑽孔座標檔。

檔案選取完畢後、系統即依據鑽孔座標檔內記錄之座標選取鑽孔內容，而後再推算出地層分層之結果，並顯示土層剖面圖於螢幕。如圖5.6。

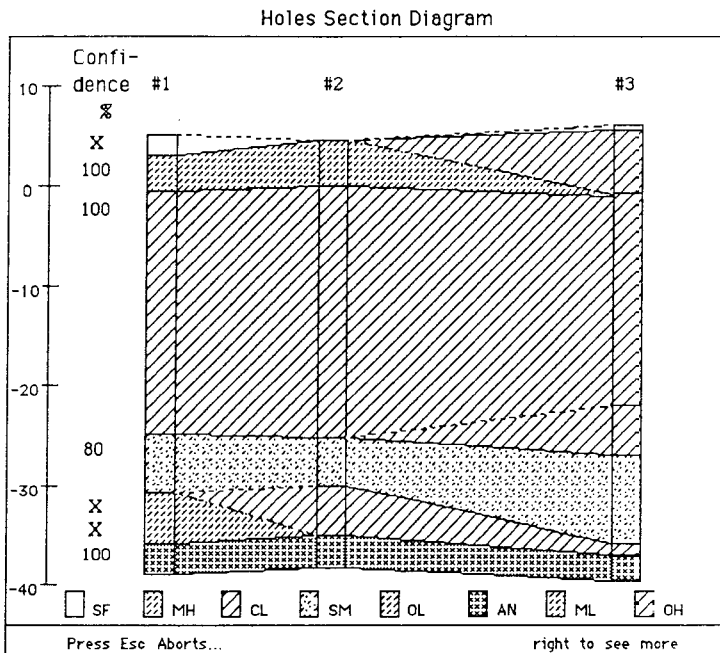


圖5.6 土層剖面圖

圖中各土層是以不同的圖案，表示不同的土壤分類，在畫面的底部即顯示圖案與土壤分類的對照表，使用者即可由對照表知道畫面上的鑽孔與推理機所推斷的土層是由那些土壤組成的。畫面左側的標尺可指示出鑽孔的深度。標尺右側的信賴度指標(Confidence)，表示其右側對應的兩鑽孔土層間配對時所依據的相似值。"X"表示土層無法配對而產生尖滅層。若剖面圖有兩個鑽孔以上，可利用左右的方向鍵觀看其他鑽孔土層匹配的情形。以圖5.6 為例，其表示了圖中鑽孔1(#1)與鑽孔2(#2)各層次匹配的信賴度，若欲查看鑽孔2(#2)與鑽孔3(#3)各層次之信賴度，則可按向右之方向鍵觀看之(圖5.7)。

Holes Section Diagram

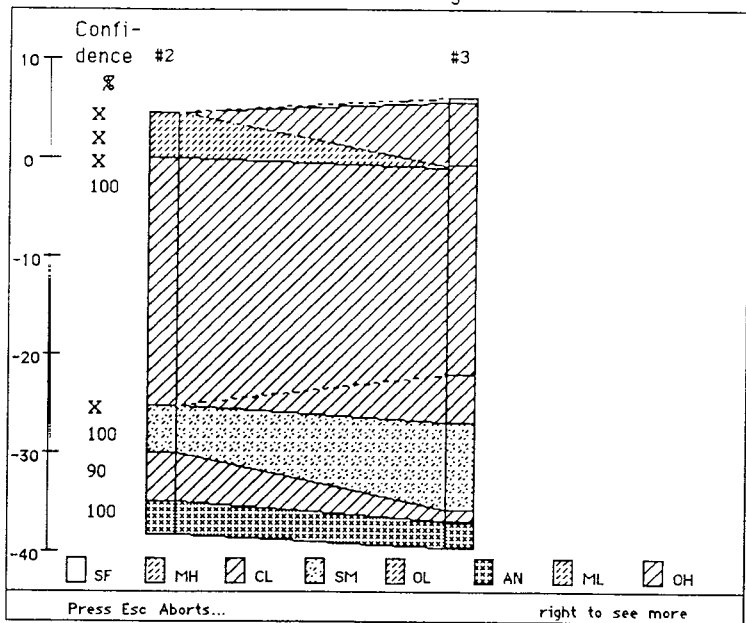


圖 5.7 土層剖面圖

假若使用者欲直接輸入其所欲分析鑽孔之座標則其可在圖 5.3之畫面選取第二項功能，系統即馬上呈現圖5.8之畫面。

Holes Coordinate

Selected Coordinate
Are ...

X :

Please Enter the X Coordinate ...

Blank : Select Over

圖 5.8 鑽孔座標輸入畫面

使用者即可依序填入 X、Y 座標，系統會檢測此座標值是否已存在於資料庫中，若此座標尚未存於資料庫中，則系統將不接受此座標，而在螢幕底部顯示警告訊息。若使用者已將所有欲分析之鑽孔座標輸入完畢，則連續按兩個Return鍵，使 X、Y座標值均為空白，如圖5.9所示：

Holes Coordinate

Selected Coordinate

Are ...

X : 301455	Y : 2778308
X : 301876	Y : 2777492
X : 301489	Y : 2778254

X : Y :

Please Enter the Y Coordinate ...

Blank : Select Over

圖 5.9 鑽孔座標輸入結束畫面

則系統便開始分析使用者所選定之鑽孔，並將分析結果送至圖形顯示界面，於螢幕上畫出土層剖面圖。

(2) 土層相似性測試系統

為了幫助土壤分層研究者能夠即時獲得土壤分層之相關資料，本專家系統擬提出三個功能以作為其相關資料試驗參考之用。

本單元係分析兩土層之相似性，其表現方式是提出兩個數值：相似值(Similarity)，相異值(Unsimilarity)，相似值與相異值之總和為100，相似值愈大則代表兩土層之相似性愈大。

當使用者在主畫面(圖5.2)選取第3項功能時，則螢幕呈現圖5.10 畫面：

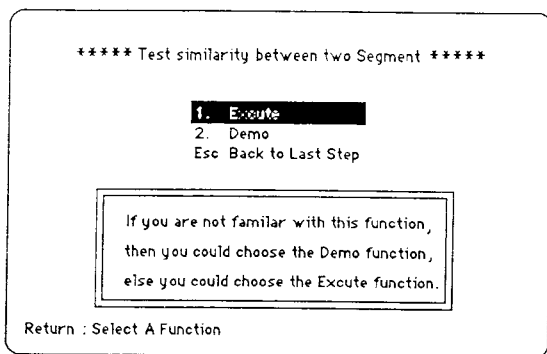


圖 5.10 測試土層相似性功能表

其中"EXCUTE"功能即讓使用者執行其欲分析之土層，而為了讓初用之使用者，了解輸入之格式，特提供了"DEMO"之功能，此功能即產生一範例教學(Tutorial)之格式讓使用者得以遵循

以圖5.11為例，使用者填入兩土層資料後，系統即根據兩土層之組成經由推論機推算出相似值與相異值。

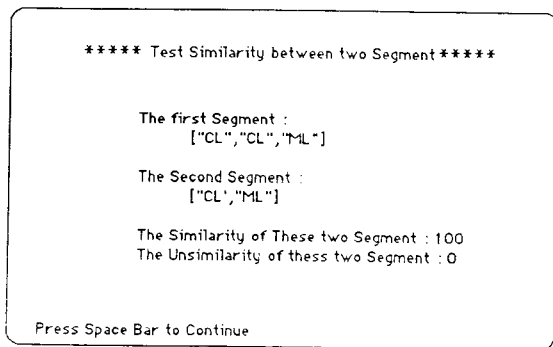


圖 5.11 土層相似性測試畫面

由此例可看出此二土層完全相似。

(3) 土層分割測試子系統

相同於土層相似性測試子系統，土層分割子系統仍然包含了'EXECUTE'與'DEMO'兩個子功能。如圖5.12之結果。

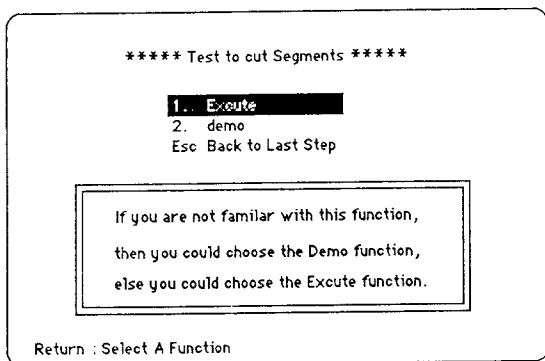


圖 5.12 土層分層功能表

本子系統係讓使用者輸入某單一土層資料而後利用專家系統推論其是否須再分割。若可再分割，則將顯示出分割後之結果。

當使用者在圖5.12螢幕選擇執行(EXCUTE)之功能後，系統將詢問使用者土層內之土岩內容與決定相似與否之參考權值。當輸入完全後，系統即根據參考權值與知識庫之資料而判斷輸入之土層是否須再予以分層，茲將結果顯示如圖5.13：

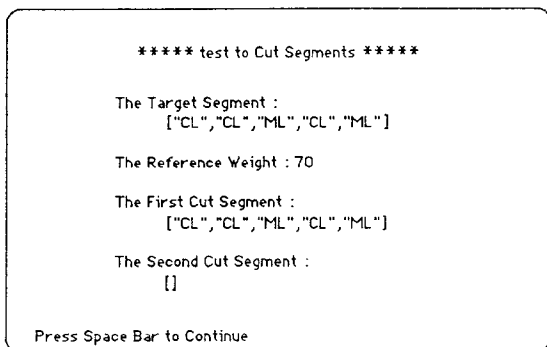


圖 5.13 土層分層測試畫面

同樣地，此子系統亦有 "DEMO" 之功能。

(4) 鑽孔比對與層次細分測試子系統

本子系統之主要目的是欲測試某鑽孔在相對於一參考鑽孔比較之下視其原先之分層是否適當，若尚須分層則推理機將以參考鑽孔為依據，而予以重新細分，並且將結果顯示出來。

當使用者在主功能螢幕選擇第 5 項功能時，系統即進入此鑽孔匹配測試子系統，螢幕即呈現圖 5.14 之形式：

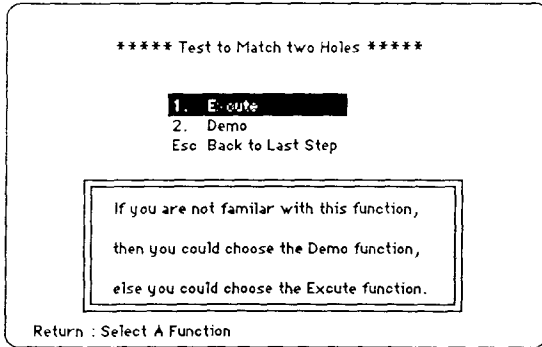


圖 5.14 鑽孔比對功能表

圖 5.15 即為一測試結果：

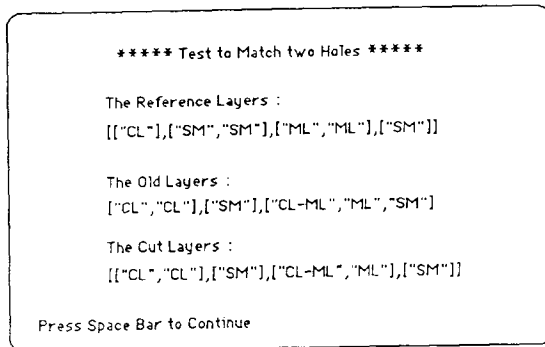


圖 5.15 鑽孔比對測試畫面

由上例可知，欲分層之土層其最後一層之"sm"被重新分割，而視為一獨立土層。

若使用者不熟悉此功能之運作情形，同樣地，可選擇"DEMO"程序學習之。

外部資料萃取模組與知識編輯模組已在第一節描述，於此不再贅述。

六、結論與建議

6.1 結論

土層剖面圖專家系統之土層匹配工作，非一般性之匹配工作，故其推論機之設計亦異於一般性推論機之設計。本研究取消Aho與Peterson之字串辨識理論中之「取代錯換」功能，但保留符合土壤分層特性之其餘錯換定義，並引用Fu之權重概念，作為分層辨識之方法；更改圖形理論之最短路徑求解過程作為土層匹配之路徑搜尋程序；以大中取小原則，作為土層是否再細分之基礎。使用邏輯語言Turbo Prolog及其內建推論機，並以自創之土壤分層辨識方法另造一外建推論機，共同進行土壤分層之推論工作。外建推論機之推論步驟可分為鄰孔之比對，層次細分，多孔層次之最佳匹配，以及土層劃分之決定等四大步驟。由本研究實際建構之系統而言，其分層之結果大致能得到不錯的效果。使用者界面之設計利用圖形表示其推論的過程與結果，而能獲致使用者相當程度的接受，可知一個親切友善的使用者界面，確實是一系統成功與否的重大因素之一。

6.2 建議

本研究推理機之推理方式，由實際運作之情形來看，大致已獲肯定。未來則可再考慮更多之經驗法則，及更深入之土／岩區分，以從事上下鄰層之合併研究：

由於大地工程地質資料庫內的土／岩分類，大多是於現場鑽探時，由現場人員以其經驗判斷所獲之樣本為何種土／岩分類，而並非依照樣本試驗後之土壤的物理性質，依照土壤分類系統推算而得。因此專家系統也可再開發一資料偵錯暨校正之功能，使其就大地工程地質資料庫所載之各鑽孔的各個物理性質，重新判定其土壤分類是否正確：

由本研究根據最小路徑搜尋策略所改寫的控制策略，也常適用於其他之應用領域，因此考慮如何將本研究之最小路徑搜尋策略改寫為一般用途 (general purpose) 的控制機制，也是一個非常值得研究的課題。

由系統推論出的土層剖面圖，主要是提供土層參考之用，因此若是當土層剖面圖顯示後能提供編輯之功能，而增加地層專家之修正意見，則最後的剖面圖才能名符其實地成為專家推論後的土層剖面圖。

附錄二 N值對等值表

〔一〕黏性土之N值對等值(知識來源：李咸亨 教授)

	極軟弱	軟弱	中等堅實	堅實	極堅實	堅硬
極軟弱	100	85	70	50	30	10
軟弱		100	85	70	50	30
中等堅實			100	85	70	50
堅實				100	85	70
極堅實					100	85
堅硬						100

〔二〕無黏性土之N值對等值(知識來源：李咸亨 教授)

	極疏鬆	疏鬆	中等緊密	緊密	極緊密
極疏鬆	100	85	70	50	30
疏鬆		100	85	70	50
中等緊密			100	85	70
緊密				100	85
極緊密					100

附錄三 土壤對等值與 N 值對等值 之知識表示

largest_weight(i00)

ref_weight(70)

eq_sub11("GW","GW-GM",95)
eq_sub11("GW","GW-GM-GC",90)
eq_sub11("GW","GW-GC",85)
eq_sub11("GW","GP",69)
eq_sub11("GW","GP-GM",68)
eq_sub11("GW","GP-GM-GC",67)
eq_sub11("GW","GP-GC",65)
eq_sub11("GW","GM",60)
eq_sub11("GW","GM-GC",50)
eq_sub11("GW","GC",50)
eq_sub11("GW-GM","GW-GM-GC",95)
eq_sub11("GW-GM","GW-GC",90)
eq_sub11("GW-GM","GP",81)
eq_sub11("GW-GM","GP-GM",81)
eq_sub11("GW-GM","GP-GM-GC",69)
eq_sub11("GW-GM","GP-GC",67)
eq_sub11("GW-GM","GM",65)
eq_sub11("GW-GM","GM-GC",52)
eq_sub11("GW-GM","GC",51)
eq_sub11("GW-GM-GC","GW-GC",95)
eq_sub11("GW-GM-GC","GP",82)
eq_sub11("GW-GM-GC","GP-GM",82)
eq_sub11("GW-GM-GC","GP-GM-GC",81)
eq_sub11("GW-GM-GC","GP-GC",69)
eq_sub11("GW-GM-GC","GM",65)
eq_sub11("GW-GM-GC","GM-GC",60)
eq_sub11("GW-GM-GC","GC",53)
eq_sub11("GW-GC","GP",83)
eq_sub11("GW-GC","GP-GM",82)
eq_sub11("GW-GC","GP-GM-GC",82)
eq_sub11("GW-GC","GP-GC",81)
eq_sub11("GW-GC","GM",65)
eq_sub11("GW-GC","GM-GC",62)

eq_sub11("GW-GC","GC",55)
eq_sub11("GP","GP-GM",95)
eq_sub11("GP","GP-GM-GC",90)
eq_sub11("GP","GP-GC",85)
eq_sub11("GP","GM",69)
eq_sub11("GP","GM-GC",65)
eq_sub11("GP","GC",60)
eq_sub11("GP-GM","GP-GM-GC",95)
eq_sub11("GP-GM","GP-GC",90)
eq_sub11("GP-GM","GM",80)
eq_sub11("GP-GM","GM-GC",69)
eq_sub11("GP-GM","GC",65)
eq_sub11("GP-GM-GC","GP-GC",95)
eq_sub11("GP-GM-GC","GM",85)
eq_sub11("GP-GM-GC","GM-GC",80)
eq_sub11("GP-GM-GC","GC",69)
eq_sub11("GP-GC","GM",85)
eq_sub11("GP-GC","GM-GC",85)
eq_sub11("GP-GC","GC",80)
eq_sub11("GM","GM-GC",95)
eq_sub11("GM","GC",90)
eq_sub11("GM-GC","GC",95)

eq_sub12("GW","SW",30)
eq_sub12("GW","SW-SM",20)
eq_sub12("GW","SW-SM-SC",20)
eq_sub12("GW","SW-SC",20)
eq_sub12("GW","SP",18)
eq_sub12("GW","SP-SM",15)
eq_sub12("GW","SP-SM-SC",10)
eq_sub12("GW","SP-SC",10)
eq_sub12("GW","SM",10)
eq_sub12("GW-GM","SW",32)
eq_sub12("GW-GM","SW-SM",22)
eq_sub12("GW-GM","SW-SM-SC",22)
eq_sub12("GW-GM","SW-SC",22)
eq_sub12("GW-GM","SP",20)
eq_sub12("GW-GM","SP-SM",16)
eq_sub12("GW-GM","SP-SM-SC",10)
eq_sub12("GW-GM","SP-SC",10)
eq_sub12("GW-GM","SM",10)
eq_sub12("GW-GM-GC","SW",34)
eq_sub12("GW-GM-GC","SW-SM",24)

eq_sub12("GW-GM-GC","SW-SM-SC",24)
eq_sub12("GW-GM-GC","SW-SC",24)
eq_sub12("GW-GM-GC","SP",22)
eq_sub12("GW-GM-GC","SP-SM",17)
eq_sub12("GW-GM-GC","SP-SM-SC",10)
eq_sub12("GW-GM-GC","SP-SC",10)
eq_sub12("GW-GM-GC","SM",10)
eq_sub12("GW-GM-GC","SM-SC",10)
eq_sub12("GW-GC","SW",34)
eq_sub12("GW-GC","SW-SM",26)
eq_sub12("GW-GC","SW-SM-SC",26)
eq_sub12("GW-GC","SW-SC",26)
eq_sub12("GW-GC","SP",24)
eq_sub12("GW-GC","SP-SM",18)
eq_sub12("GW-GC","SP-SM-SC",10)
eq_sub12("GW-GC","SP-SC",10)
eq_sub12("GW-GC","SM",10)
eq_sub12("GW-GC","SM-SC",10)
eq_sub12("GP","SW",40)
eq_sub12("GP","SW-SM",30)
eq_sub12("GP","SW-SM-SC",30)
eq_sub12("GP","SW-SC",30)
eq_sub12("GP","SP",28)
eq_sub12("GP","SP-SM",25)
eq_sub12("GP","SP-SM-SC",20)
eq_sub12("GP","SP-SC",10)
eq_sub12("GP","SM",10)
eq_sub12("GP","SM-SC",10)
eq_sub12("GP","SC",10)
eq_sub12("GP-GM","SW",42)
eq_sub12("GP-GM","SW-SM",32)
eq_sub12("GP-GM","SW-SM-SC",31)
eq_sub12("GP-GM","SW-SC",31)
eq_sub12("GP-GM","SP",30)
eq_sub12("GP-GM","SP-SM",27)
eq_sub12("GP-GM","SP-SM-SC",20)
eq_sub12("GP-GM","SP-SC",15)
eq_sub12("GP-GM","SM",10)
eq_sub12("GP-GM","SM-SC",10)
eq_sub12("GP-GM","SC",10)
eq_sub12("GP-GM-GC","SW",44)
eq_sub12("GP-GM-GC","SW-SM",34)
eq_sub12("GP-GM-GC","SW-SM-SC",32)

eq_sub12("GP-GM-GC","SW-SC",32)
eq_sub12("GP-GM-GC","SP",31)
eq_sub12("GP-GM-GC","SP-SM",30)
eq_sub12("GP-GM-GC","SP-SM-SC",25)
eq_sub12("GP-GM-GC","SP-SC",20)
eq_sub12("GP-GM-GC","SM",15)
eq_sub12("GP-GM-GC","SM-SC",10)
eq_sub12("GP-GM-GC","SC",10)
eq_sub12("GP-GC","SW",44)
eq_sub12("GP-GC","SW-SM",34)
eq_sub12("GP-GC","SW-SM-SC",33)
eq_sub12("GP-GC","SW-SC",33)
eq_sub12("GP-GC","SP",32)
eq_sub12("GP-GC","SP-SM",30)
eq_sub12("GP-GC","SP-SM-SC",25)
eq_sub12("GP-GC","SP-SC",20)
eq_sub12("GP-GC","SM",15)
eq_sub12("GP-GC","SM-SC",10)
eq_sub12("GP-GC","SC",10)
eq_sub12("GM","SW",45)
eq_sub12("GM","SW-SM",40)
eq_sub12("GM","SW-SM-SC",35)
eq_sub12("GM","SW-SC",33)
eq_sub12("GM","SP",33)
eq_sub12("GM","SP-SM",30)
eq_sub12("GM","SP-SM-SC",26)
eq_sub12("GM","SP-SC",22)
eq_sub12("GM","SM",17)
eq_sub12("GM","SM-SC",10)
eq_sub12("GM","SC",10)
eq_sub12("GM-GC","SW",47)
eq_sub12("GM-GC","SW-SM",45)
eq_sub12("GM-GC","SW-SM-SC",40)
eq_sub12("GM-GC","SW-SC",35)
eq_sub12("GM-GC","SP",33)
eq_sub12("GM-GC","SP-SM",31)
eq_sub12("GM-GC","SP-SM-SC",27)
eq_sub12("GM-GC","SP-SC",23)
eq_sub12("GM-GC","SM",18)
eq_sub12("GM-GC","SM-SC",10)
eq_sub12("GM-GC","SC",10)
eq_sub12("GC","SW",49)
eq_sub12("GC","SW-SM",47)

eq_sub12("GC","SW-SM-SC",40)
eq_sub12("GC","SW-SC",40)
eq_sub12("GC","SP",35)
eq_sub12("GC","SP-SM",33)
eq_sub12("GC","SP-SM-SC",28)
eq_sub12("GC","SP-SC",25)
eq_sub12("GC","SM",20)
eq_sub12("GC","SM-SC",15)
eq_sub12("GC","SC",10)

eq_sub13("GP-GM-GC","ML",1)
eq_sub13("GP-GC","ML",2)
eq_sub13("GP-GC","CL-ML",1)
eq_sub13("GM","ML",3)
eq_sub13("GM","CL-ML",1)
eq_sub13("GM-GC","ML",4)
eq_sub13("GM-GC","CL-ML",2)
eq_sub13("GM-GC","CL",1)
eq_sub13("GC","ML",5)
eq_sub13("GC","CL-ML",3)
eq_sub13("GC","CL",1)

eq_sub22("SW","SW-SM",90)
eq_sub22("SW","SW-SM-SC",80)
eq_sub22("SW","SW-SC",70)
eq_sub22("SW","SP",60)
eq_sub22("SW","SP-SM",50)
eq_sub22("SW","SP-SM-SC",45)
eq_sub22("SW","SP-SC",40)
eq_sub22("SW","SM",30)
eq_sub22("SW","SM-SC",20)
eq_sub22("SW","SC",15)
eq_sub22("SW-SM","SW-SM-SC",90)
eq_sub22("SW-SM","SW-SC",80)
eq_sub22("SW-SM","SP",70)
eq_sub22("SW-SM","SP-SM",60)
eq_sub22("SW-SM","SP-SM-SC",50)
eq_sub22("SW-SM","SP-SC",40)
eq_sub22("SW-SM","SM",30)
eq_sub22("SW-SM","SM-SC",25)
eq_sub22("SW-SM","SC",20)
eq_sub22("SW-SM-SC","SW-SC",90)
eq_sub22("SW-SM-SC","SP",80)

eq_sub22("SW-SM-SC", "SP-SM", 70)
eq_sub22("SW-SM-SC", "SP-SM-SC", 60)
eq_sub22("SW-SM-SC", "SP-SC", 50)
eq_sub22("SW-SM-SC", "SM", 40)
eq_sub22("SW-SM-SC", "SM-SC", 30)
eq_sub22("SW-SM-SC", "SC", 25)
eq_sub22("SW-SC", "SP", 85)
eq_sub22("SW-SC", "SP-SM", 80)
eq_sub22("SW-SC", "SP-SM-SC", 70)
eq_sub22("SW-SC", "SP-SC", 60)
eq_sub22("SW-SC", "SM", 50)
eq_sub22("SW-SC", "SM-SC", 40)
eq_sub22("SW-SC", "SC", 30)
eq_sub22("SP", "SP-SM", 90)
eq_sub22("SP", "SP-SM-SC", 80)
eq_sub22("SP", "SP-SC", 70)
eq_sub22("SP", "SM", 60)
eq_sub22("SP", "SM-SC", 50)
eq_sub22("SP", "SC", 40)
eq_sub22("SP-SM", "SP-SM-SC", 90)
eq_sub22("SP-SM", "SP-SC", 80)
eq_sub22("SP-SM", "SM", 70)
eq_sub22("SP-SM", "SM-SC", 60)
eq_sub22("SP-SM", "SC", 50)
eq_sub22("SP-SM-SC", "SP-SC", 90)
eq_sub22("SP-SM-SC", "SM", 80)
eq_sub22("SP-SM-SC", "SM-SC", 70)
eq_sub22("SP-SM-SC", "SC", 60)
eq_sub22("SP-SC", "SM", 85)
eq_sub22("SP-SC", "SM-SC", 80)
eq_sub22("SP-SC", "SC", 70)
eq_sub22("SM", "SM-SC", 90)
eq_sub22("SM", "SC", 80)
eq_sub22("SM-SC", "SC", 90)

eq_sub23("SW", "ML", 10)
eq_sub23("SW", "CL-ML", 5)
eq_sub23("SW", "CL", 2)
eq_sub23("SW-SM", "ML", 12)
eq_sub23("SW-SM", "CL-ML", 8)
eq_sub23("SW-SM", "CL", 4)
eq_sub23("SW-SM", "OL", 1)
eq_sub23("SW-SM-SC", "ML", 16)

eq_sub23("SW-SM-SC","CL-ML",10)
eq_sub23("SW-SM-SC","CL",6)
eq_sub23("SW-SM-SC","OL",3)
eq_sub23("SW-SC","ML",20)
eq_sub23("SW-SC","CL-ML",12)
eq_sub23("SW-SC","CL",8)
eq_sub23("SW-SC","OL",4)
eq_sub23("SP","ML",30)
eq_sub23("SP","CL-ML",20)
eq_sub23("SP","CL",10)
eq_sub23("SP","OL",50)
eq_sub23("SP-SM","ML",40)
eq_sub23("SP-SM","CL-ML",30)
eq_sub23("SP-SM","CL",20)
eq_sub23("SP-SM","OL",8)
eq_sub23("SP-SM","MH",2)
eq_sub23("SP-SM-SC","ML",50)
eq_sub23("SP-SM-SC","CL-ML",40)
eq_sub23("SP-SM-SC","CL",25)
eq_sub23("SP-SM-SC","OL",15)
eq_sub23("SP-SM-SC","MH",5)
eq_sub23("SP-SM-SC","CH",1)
eq_sub23("SP-SC","ML",60)
eq_sub23("SP-SC","CL-ML",50)
eq_sub23("SP-SC","CL",30)
eq_sub23("SP-SC","OL",20)
eq_sub23("SP-SC","MH",10)
eq_sub23("SP-SC","CH",2)
eq_sub23("SM","ML",67)
eq_sub23("SM","CL-ML",60)
eq_sub23("SM","CL",50)
eq_sub23("SM","OL",30)
eq_sub23("SM","MH",15)
eq_sub23("SM","CH",5)
eq_sub23("SM-SC","ML",68)
eq_sub23("SM-SC","CL-ML",65)
eq_sub23("SM-SC","CL",50)
eq_sub23("SM-SC","OL",35)
eq_sub23("SM-SC","MH",25)
eq_sub23("SM-SC","CH",12)
eq_sub23("SC","ML",69)
eq_sub23("SC","CL-ML",67)
eq_sub23("SC","CL",60)

eq_sub23("SC","OL",40)
eq_sub23("SC","MH",30)
eq_sub23("SC","CH",18)
eq_sub23("SC","OH",12)

eq_sub33("ML","CL-ML",90)
eq_sub33("ML","CL",70)
eq_sub33("ML","OL",60)
eq_sub33("ML","MH",75)
eq_sub33("ML","CH",20)
eq_sub33("ML","OH",15)
eq_sub33("ML","PT",10)
eq_sub33("CL-ML","CL",85)
eq_sub33("CL-ML","OL",70)
eq_sub33("CL-ML","MH",60)
eq_sub33("CL-ML","CH",40)
eq_sub33("CL-ML","OH",20)
eq_sub33("CL-ML","PT",10)
eq_sub33("CL","OL",80)
eq_sub33("CL","MH",68)
eq_sub33("CL","CH",50)
eq_sub33("CL","OH",30)
eq_sub33("CL","PT",10)
eq_sub33("OL","MH",75)
eq_sub33("OL","CH",60)
eq_sub33("OL","OH",40)
eq_sub33("OL","PT",20)
eq_sub33("MH","CH",80)
eq_sub33("MH","OH",75)
eq_sub33("MH","PT",40)
eq_sub33("CH","OH",80)
eq_sub33("CH","PT",60)
eq_sub33("OH","PT",80)

eq_sub_n1(2,4,85)
eq_sub_n1(4,10,85)
eq_sub_n1(10,30,85)
eq_sub_n1(30,50,85)
eq_sub_n1(2,10,70)
eq_sub_n1(4,30,70)
eq_sub_n1(10,50,70)
eq_sub_n1(2,30,50)
eq_sub_n1(4,50,50)

eq_sub_n1(2,50,30)

eq_sub_n2(1,2,85)

eq_sub_n2(2,4,85)

eq_sub_n2(4,8,85)

eq_sub_n2(8,15,85)

eq_sub_n2(15,30,85)

eq_sub_n2(1,4,70)

eq_sub_n2(2,8,70)

eq_sub_n2(4,15,70)

eq_sub_n2(15,30,70)

eq_sub_n2(1,8,50)

eq_sub_n2(2,15,50)

eq_sub_n2(4,30,50)

eq_sub_n2(1,15,30)

eq_sub_n2(2,30,30)

附錄四 土層剖面圖專家系統推理機

```

/*****
      basic support
*****/
predicates
  Clear_cuttingRec
  Clear_cutedRec
  Clear_temp_lr_layers_content_of_hole
  Append_layers(Layers,Layers,Layers)      %  append_layers_of_a_hole
  Append_segments(Layer,Layer,Layer)      %  append_segments_of_a_layer
  Pack(Layer,temp_layer)
  Input_cuttingRec(Layers,integer,integer)
  Input_cutedRec(Layer,Layer,integer)

clauses
  Clear_cuttingRec:-
    Retractall(cuttingRec(_,_,_)).

  Clear_cutedRec:-
    Retractall(cutedRec(_,_,_)).

  Clear_temp_lr_layers_content_of_hole:-
    Retractall(temp_lr_layers_content_of_hole(_,_,_,_,_)).

  Append_layers([],L2,L2).
  Append_layers([X|L1],L2,[X|L3]) :-
    Append_layers(L1,L2,L3).

  Append_segments([],L2,L2).
  Append_segments([X|L1],L2,[X|L3]) :-
    Append_segments(L1,L2,L3).

  Pack([segment(X,N_Value,_,_),[temp_segment(X,N_Value)]]):-!.
  Pack([segment(X,N_Value,_,_),segment(X,N_Value,_,_)|Oldset],Newset):-
    Pack([segment(X,N_Value,_,_)|Oldset],Newset),!.
  Pack([segment(X,N_Value,_,_)|Oldset],[temp_segment(X,N_Value)|Newset]]:-
    Pack(Oldset,Newset).

  Input_cuttingRec(CuttingHole,Dist,Same):-
    Clear_cuttingRec,
    Assert(cuttingRec(CuttingHole,Dist,Same)).

```

```

Input_cutedRec(Segment1,Segment2,Simi):-
  Clear_cutedRec,
  Assert(cutedRec(Segment1,Segment2,Simi)).

/*****
                                meta similar
*****/
predicates
  Similar(Layer,Layer,integer,integer)
  Similarto(temp_layer,temp_layer,integer,integer)
  Accumulate_Weight_of_Intersect(temp_layer,temp_layer,integer)
  MemberWeight(temp_segment,temp_layer,integer)
  Count_segment_number(temp_layer,integer)
  Count_segment_number(Layer,integer)

clauses
  Similar(Layer1,Layer2,Simi,Unsimi):-
    Pack(Layer1,PackedLayer1),
    Pack(Layer2,PackedLayer2),
    Similarto(PackedLayer1,PackedLayer2,Simi,Unsimi).

  Similarto(Layer1,Layer2,Simi,Unsimi):-
    Accumulate_Weight_of_Intersect(Layer1,Layer2,Accum_Simi),
    Count_segment_number(Layer1,N1),
    Simi = round(Accum_Simi / N1),
    largest_weight(Largest_weight),
    Unsimi= Largest_weight - Simi.

  Accumulate_Weight_of_Intersect([],_,0):-!.
  Accumulate_Weight_of_Intersect([X|L1],L,Weight):-
    MemberWeight(X,L,Accum_Weight1),
    Count_segment_number(L,L_No),
    Weight1 = Accum_Weight1 Div L_No,!,
    Accumulate_Weight_of_Intersect(L1,L,Weight2),
    Weight = Weight1+Weight2.

  MemberWeight(_,[],0):-!.
  MemberWeight(X,[Y|Tail],Weight):-
    Eq(X,Y,Weight1),!,
    MemberWeight(X,Tail,Weight2),
    Weight=Weight1+Weight2,!.

```

```

Count_segment_number([],0):-!.
Count_segment_number([_:_L1],C):-
    Count_segment_number(L1,CC),
    C = CC + 1.

```

```

/*****
    evaluate difference and samevalue of a hole by reference hole
*****/

```

predicates

```

Evaluate_difference_and_samevalue_of_hole(Layers,Layers,
                                           integer,integer)

```

```

Overlength_of_hole(Layers,integer)

```

```

SimilarHorizon(Layer,Layer,integer,integer)

```

```

SimilarCriteria(integer,integer)

```

```

Dcid_Dist_Same(integer,integer,integer,integer,integer,integer)

```

clauses

```

Evaluate_difference_and_samevalue_of_hole([],Hole2,Dist,0) :-
    Overlength_of_hole(Hole2,Dist),!.

```

```

Evaluate_difference_and_samevalue_of_hole(Hole1,[],Dist,0) :-
    Overlength_of_hole(Hole1,Dist),!.

```

```

Evaluate_difference_and_samevalue_of_hole([X|Hole1],
                                           [Y|Hole2],Dist,Same) :-

```

```

    SimilarHorizon(X,Y,Simi,Unsimi),

```

```

    ref_weight(Ref_weight),

```

```

    SimilarCriteria(Simi,Ref_weight),!,

```

```

    Evaluate_difference_and_samevalue_of_hole(Hole1,Hole2,Dist1,Same1),

```

```

    Dist = Dist1 + Unsimi,

```

```

    Same = Same1 + Simi,!.

```

```

Evaluate_difference_and_samevalue_of_hole([X|Hole1],
                                           [Y|Hole2],Dist,Same) :-

```

```

    Evaluate_difference_and_samevalue_of_hole([X|Hole1],

```

```

                                           Hole2,Dist1,Same1 ),!,

```

```

    Evaluate_difference_and_samevalue_of_hole(Hole1,

```

```

                                           [Y|Hole2],Dist2,Same2),!,

```

```

    Dcid_Dist_Same(Dist1,Dist2,Dist,Same1,Same2,Same).

```

```

Overlength_of_hole([],0):-!.

```

```

Overlength_of_hole([_:_L1],Dist):-

```

```

    Overlength_of_hole(L1,Dist1),

```

```

    largest_weight(Largest_weight),

```

```

    Dist = Dist1 + Largest_weight.

```



```
SimilarHorizon(Layer1,Layer2,Simi,Unsimi):-  
    Similar(Layer1,Layer2,Simi,Unsimi).
```

```
SimilarCriteria(Simi,Reference_Weight):-  
    Simi >= Reference_Weight.
```

```
Dcid_Dist_Same(Dist1,Dist2,Dist,Same1,Same2,Same):-  
    Dist1 - Dist2 > Same1 - Same2,  
    largest_weight(Largest_weight),  
    Dist = Largest_weight + Dist2,  
    Same = Same2,!.  
Dcid_Dist_Same(Dist1,_,Dist,Same1,_,Same):-  
    largest_weight(Largest_weight),  
    Dist = Largest_weight + Dist1,  
    Same = Same1.
```

```
/******  
Create Layer Pairs between Matching Holes  
******/
```

```
predicates
```

```
    Create_a_layerpair(Layers,real,Layers,real,Layer_pairs,integer,integer)  
    Rest_of_righthole(real,Layers,Layer_pairs,integer)  
    Rest_of_lefthole(real,Layers,Layer_pairs,integer)  
    Dcid_HolePair(Layer,Layer,Layer_pairs,Layer_pairs,  
        Layer_pairs,integer,integer,integer,integer,integer)  
    Get_depth_of_a_layer(Layer,string,real,real)  
    Get_class_of_a_layer(Layer,string)  
    Get_begin_of_a_layer(Layer,real)  
    Get_end_of_a_layer(Layer,real)  
    Rearrange_segments(Layer,Layer)
```

```
clauses
```

```
    Create_a_layerpair([],End1,Right_Hole,_,HolePairs,Dist,0):-  
        Rest_of_righthole(End1,Right_Hole,HolePairs,Dist),!.  
    Create_a_layerpair(Left_Hole,_,[],End2,HolePairs,Dist,0):-  
        Rest_of_lefthole(End2,Left_Hole,HolePairs,Dist),!.  
    Create_a_layerpair([Layer_of_lefthole|Left_Hole],_,  
        [Layer_of_righthole|Right_Hole],_,  
        [layer_pair(Class,Conf,depth(Begin1,End1),  
            depth(Begin2,End2))|Pairs],  
        Dist, Same):-  
        SimilarHorizon(Layer_of_lefthole,Layer_of_righthole,Simi,Unsimi),
```

```

ref_weight(Ref_weight),
SimilarCriteria(Simi,Ref_weight),
  Get_depth_of_a_layer(Layer_of_lefthole,Class,Begin1,End1),
  Get_depth_of_a_layer(Layer_of_righthole,_,Begin2,End2),
  Create_a_layerpair(Left_Hole,End1,Right_Hole,End2,Pairs,Dist1,Same1),
Str_int(Conf,Simi),
Dist = Dist1 + Unsimi,
Same = Same1 + Simi,!.
Create_a_layerpair([Layer_of_lefthole|Left_Hole],_,
  [Layer_of_righthole|Right_Hole],_,
  [layer_pair(Class,Conf,depth(Begin1,End1),
    depth(Begin2,Begin2))|Pairs],
    Dist, Same):-
  Get_depth_of_a_layer(Layer_of_lefthole,Class,Begin1,End1),
  Get_depth_of_a_layer(Layer_of_righthole,_,Begin2,_),
Class="SF",
  Create_a_layerpair(Left_Hole,End1,
    [Layer_of_righthole|Right_Hole],_,Pairs,Dist1,Same),
Conf = "X",
  largest_weight(Largest_weight),
  Dist = Dist1 + Largest_weight,!.
Create_a_layerpair([Layer_of_lefthole|Left_Hole],_,
  [Layer_of_righthole|Right_Hole],_,
  [layer_pair(Class,Conf,depth(Begin1,Begin1),
    depth(Begin2,End2))|Pairs],
    Dist, Same):-
  Get_depth_of_a_layer(Layer_of_lefthole,_,Begin1,_),
  Get_depth_of_a_layer(Layer_of_righthole,Class,Begin2,End2),
Class="SF",
  Create_a_layerpair([Layer_of_lefthole|Left_Hole],_,
    Right_Hole,End2,Pairs,Dist1,Same),
Conf = "X",
  largest_weight(Largest_weight),
  Dist = Dist1 + Largest_weight,!.
Create_a_layerpair([Layer_of_lefthole|Left_Hole],_,
  [Layer_of_righthole|Right_Hole],_,
    HolePairs,Dist,Same):-
  Get_depth_of_a_layer(Layer_of_lefthole,_,_,End1),
  Get_depth_of_a_layer(Layer_of_righthole,_,_,End2),
  Create_a_layerpair([Layer_of_lefthole|Left_Hole],_,Right_Hole,End2,
    HolePairsTemp1,Dist1,Same1),
  Create_a_layerpair(Left_Hole,End1,[Layer_of_righthole|Right_Hole],_,
    HolePairsTemp2,Dist2,Same2),

```

```

Dcid_HolePair(Layer_of_lefthole,Layer_of_righthole,
              HolePairsTemp1,HolePairsTemp2,HolePairs,
              Dist1,Dist2,Dist,Same1,Same2,Same).

```

```

Rest_of_righthole(_,[],[],0):-!.
Rest_of_righthole(End1,[Layer|Right_Hole],
                  [layer_pair(X,Conf,depth(End1,End1),
                              depth(Begin,End))|HolePairs],Dist):-
    Get_depth_of_a_layer(Layer,X,Begin,End),
    Rest_of_righthole(End1,Right_Hole,HolePairs,Dist1),
    Conf="X",
    largest_weight(Largest_weight),
    Dist = Dist1 + Largest_weight.

```

```

Rest_of_lefthole(_,[],[],0):-!.
Rest_of_lefthole(End2,[Layer|Left_Hole],
                 [layer_pair(Class,Conf,depth(Begin,End),
                                     depth(End2,End2))|HolePairs],Dist):-
    Get_depth_of_a_layer(Layer,Class,Begin,End),
    Rest_of_lefthole(End2,Left_Hole,HolePairs,Dist2),
    Conf="X",
    largest_weight(Largest_weight),
    Dist = Dist2 + Largest_weight.

```

```

Dcid_HolePair([X|X1],[segment(____,Begin2,_)|_],_,HolePairTemp2,
              HolePair,Dist1,Dist2,Dist,Same1,Same2,Same):-
    Dist1 - Dist2 > Same1 - Same2,
    Get_depth_of_a_layer([X|X1],Class,Begin1,End1),
    largest_weight(Largest_weight),
    Dist = Dist2 + Largest_weight,
    Same = Same2,
    Conf="X",
    HolePair = [layer_pair(Class,Conf,depth(Begin1,End1),
                          depth(Begin2,Begin2))|HolePairTemp2],!.

```

```

Dcid_HolePair([segment(____,Begin1,_)|_],[Y|Y1],
              HolePairTemp1,_,HolePair,Dist1,_,Dist,Same1,_,Same):-
    Get_depth_of_a_layer([Y|Y1],Class,Begin2,End2),
    largest_weight(Largest_weight),
    Dist = Dist1 + Largest_weight,
    Same = Same1,
    Conf="X",
    HolePair = [layer_pair(Class,Conf,depth(Begin1,Begin1),
                          depth(Begin2,End2))|HolePairTemp1].

```

```

Get_depth_of_a_layer(X,Class,Begin,End):-
  Get_class_of_a_layer(X,Class),
  Get_begin_of_a_layer(X,Begin),
  Rearrange_segments(X,Temp_X),
  Get_end_of_a_layer(Temp_X,End).

Get_class_of_a_layer([segment(Class,_,_)|_],Class).

Get_begin_of_a_layer([segment(_,Begin,_)|_],Begin).

Get_end_of_a_layer([segment(_,_,End)|_],End).

Rearrange_segments([],[]):-!.
Rearrange_segments([X1|X],Temp):-
  Rearrange_segments(X,Temp1),
  Append_layer_and_segment(Temp1,X1,Temp).

/*****
      kb goodcut for cut segment
*****/
predicates
  BestCut(Layer,Layer,integer)      % cut if both segments are not similar
  SimilarVertical(Layer,Layer,integer,integer)
  CutCriteria(integer,integer)

clauses
  BestCut(Segment1,Segment2,Reference_Weight):-
    SimilarVertical(Segment1,Segment2,Simi,_),
    CutCriteria(Simi,Reference_Weight),
    cutedRec(_,_,OldSimi),
    Simi < OldSimi,
    Input_cutedRec(Segment1,Segment2,Simi).

  SimilarVertical(Segment1,Segment2,Simi,Unsimi):-
    Similar(Segment1,Segment2,Simi,Unsimi).

  CutCriteria(Simi,Reference_Weight):-
    Simi < Reference_Weight.

```

```

/*****
  bestf finds the best-fited splitting of CutingRecord, named Last,
  ref.to L recursively using justhole
*****/

```

predicates

```

Try_to_make_a_bestfitting(Layers,Layers)
Bestfit(Layers,Layers,Layers,Layers,integer,integer)
Fithole(Layers,Layers,Layers,integer,integer)
Repeat_fit_segments(Layers,Layers)
Fitsegment(Layers,Layers,Layer,Layers)
Test_if_fit_the_reference_hole(Layers,Layers,Layers)
Try_to_fit(Layers,Layers)
Try_to_cut_segment(Layer,integer)
Check_if_pure_layer(temp_layer)

```

clauses

```

Try_to_make_a_bestfitting(RefHole,CutedHole):-
  Bestfit(RefHole,_,CutedHole,_,_).

Bestfit(RefHole,OldHole,CutedHole,NewHole,OldDist,OldSame):-
  cuttingRec(OldHole,OldDist,OldSame),
  Fithole(RefHole,OldHole,NewHole,NewDist,NewSame),
  NewSame - OldSame <= NewDist - OldDist ,
  CutedHole = OldHole,!.
Bestfit(RefHole,OldHole,CutedHole,NewHole,OldDist,OldSame):-
  % cut more segments
  CuttingRec(OldHole,OldDist,OldSame),
  Fithole(RefHole,OldHole,NewHole,_,_),
  Bestfit(RefHole,OldHole,CutedHole,NewHole,OldDist,OldSame).

Fithole(RefHole,OldHole,NewHole,OldDist,Same):-
  Repeat_fit_segments(RefHole,OldHole),
  cuttingRec(NewHole,OldDist,Same).

Repeat_fit_segments(RefHole,OldHole):-
  Append_layers(Up,[Mid_layer|Bottom],OldHole),
  Pack(Mid_layer,Packed_layer),
  Not(check_if_pure_layer(Packed_layer)),
  Fitsegment(RefHole,Up,Mid_layer,Bottom),
  Fail.
Repeat_fit_segments(,_).

```

```

Fitsegment(RefHole,Up,Mid_layer,Bottom):-
    largest_weight(Largest_weight),
    Input_cuttedRec(Mid_layer,[],Largest_weight),
    ref_weight(Ref_weight),
    Try_to_cut_segment(Mid_layer,Ref_weight),
    Test_if_fit_the_reference_hole(RefHole,Up,Bottom),!.
Fitsegment(_,_,_).

Test_if_fit_the_reference_hole(RefHole,Up,Bottom):-
    cuttedRec(X1,X2,_),
    Append_layers(Up,[X1,X2|Bottom],HoleTemp),
    Try_to_fit(RefHole,HoleTemp).

Try_to_fit(RefHole,HoleTemp):-

Evaluate_difference_and_samevalue_of_hole(RefHole,HoleTemp,Dist,Same),
    cuttingRec(_,OldDist,OldSame),
    Same - OldSame >= Dist - OldDist,
    Input_cuttingRec(HoleTemp,Dist,Same).

Try_to_cut_segment([],_):-!,fail.
Try_to_cut_segment(Segments,Reference_Weight):-
    Append_segments(Segments1,Segments2,Segments),
    Count_segment_number(Segments1,C1), C1>0,
    Count_segment_number(Segments2,C2), C2>0,
    BestCut(Segments1,Segments2,Reference_Weight),
    Fail.
Try_to_cut_segment(_,_).

Check_if_pure_layer([_]).           % if pure layer then means need not cut

/*****
    matching_holes_from_left scan and modify hole one by one increasedly
    matching_holes_from_right scan and modify hole one by one decreasedly
*****/
predicates
    Matching_hole_from_left
    Matching_hole_from_right
    Next_right_hole(integer,Layers)
    Next_left_hole(integer,Layers)

```

clauses

```
Matching_hole_from_left:-
  temp_lr_layers_content_of_hole(No,CordX,CordY,Begin,End,CutingHole),
  Next_right_hole(No,RefHole),
  Evaluate_difference_and_samevalue_of_hole(CutingHole,
                                             RefHole,Dist,Same),
  Input_cuttingRec(CutingHole,Dist,Same),
  Try_to_make_a_bestfitting(RefHole,CutedHole),
  Asserta(temp_lr_layers_content_of_hole(No,CordX,CordY,
                                         Begin,End,CutedHole)),
Fail.
Matching_hole_from_left:-!.
```

```
Matching_hole_from_right:-
  temp_rl_layers_content_of_hole(No,CordX,CordY,Begin,End,CutingHole),
  Next_left_hole(No,RefHole),
  Evaluate_difference_and_samevalue_of_hole(CutingHole,
                                             RefHole,Dist,Same),
  Input_cuttingRec(CutingHole,Dist,Same),
  Try_to_make_a_bestfitting(RefHole,CutedHole),
  Asserta(temp_rl_layers_content_of_hole(No,CordX,CordY,
                                         Begin,End,CutedHole)),
Fail.
Matching_hole_from_right:-!.
```

```
Next_right_hole(No,Hole):-
  NextNo = No + 1,
  temp_lr_layers_content_of_hole(NextNo,_,_,_,Hole),!.
Next_right_hole(No,_):-
  temp_lr_layers_content_of_hole(No,CordX,CordY,Begin,End,Hole),
  Asserta(temp_rl_layers_content_of_hole(No,CordX,CordY,
                                         Begin,End,Hole)),
  Clear_temp_lr_layers_content_of_hole,!.
Fail.
```

```
Next_left_hole(No,Hole):-
  LastNo = No - 1,
  temp_rl_layers_content_of_hole(LastNo,_,_,_,Hole),!.
Next_left_hole(No,_):-
  temp_rl_layers_content_of_hole(No,CordX,CordY,Begin,End,Hole),
  Asserta(temp_lr_layers_content_of_hole(No,CordX,CordY,
                                         Begin,End,Hole)),!.
Fail.
```

```

/*****
testing procedure
*****/
predicates
  Testsimi(Layer,Layer,integer,integer)
  Testcutsegment(Layer,Layer,Layer,integer,integer)
  Testjusthole(Layers,Layers,Layers)
  Cutsegmenttemp(Layer,integer)           % just for testcutsegment

clauses
  Testsimi(Segment1,Segment2,Simi,Unsimi):-
    Retractall(_,similar_kba),
    Retractall(_,holedata),
    Consult("fact.kba",similar_kba),
    Similar(Segment1,Segment2,Simi,Unsimi).

  Testcutsegment(Segment,Segment1,Segment2,Reference_Weight):-
    Retractall(_,similar_kba),
    Retractall(_,holedata),
    Consult("fact.kba",similar_kba),
    Cutsegmenttemp(Segment,Reference_Weight),
    cutedRec(Segment1,Segment2,_),
    Clear_cutedRec.

  Testjusthole(RefHole,OldHole,NewHole):-
    Retractall(_,similar_kba),
    Retractall(_,holedata),
    Consult("fact.kba",similar_kba),
    Evaluate_difference_and_samevalue_of_hole(RefHole,OldHole,Dist,Same),
    Input_cutingRec(OldHole,Dist,Same),
    Try_to_make_a_bestfitting(RefHole,NewHole),!.
  Testjusthole(_,_,[]).

  Cutsegmenttemp([],_):-!,fail.           /* just for testcutsegment */
  Cutsegmenttemp([_],_):-!,fail.
  Cutsegmenttemp(Segment,Reference_Weight):-
    largest_weight(Largest_weight),
    Input_cutedRec(Segment,[],Largest_weight),
    Try_to_cut_segment(Segment,Reference_Weight).

```



```

/*****
Inference Engine's Body
*****/

```

predicates

```

Bestfit_decision(Hole_IDs)
Catch_layers_content_and_depth_of_holes(Hole_IDs)
Find_layers_content_and_depth_of_holes(integer,Hole_IDs)
Create_layerpairs
Next_temp_layers_content_of_hole(integer,Layers)

```

clauses

```

Bestfit_decision(Hole_IDs):-
    Draw_architecture,
    Setcolor(white),
    StatusLine(" I am Thinking to Divide the Layers ..."),
    % defined in draw.pro

    Consult("fact.kba",similar_kba),
    Retractall(_,holedata),
    Catch_layers_content_and_depth_of_holes(Hole_IDs),
    Matching_hole_from_left,
    Matching_hole_from_right,
    Create_layerpairs,
    Retractall(_,similar_kba),
    Setbkcolor(black),
    Closegraph,!

```

```

Catch_layers_content_and_depth_of_holes(Hole_IDs):-
    Db_open(f_hole,"hole.dba",in_file),
    Find_layers_content_and_depth_of_holes(1,Hole_IDs),
    Db_close(f_hole).

```

```

Find_layers_content_and_depth_of_holes(_,[]):-!.
Find_layers_content_and_depth_of_holes(No,[Hole_ID!Holes_ID]):-
    Chain_terms(f_hole,c_hole,hole_dom,
        hole(Hole_ID,
            hole_content(Begin,End,Layers)),_),
    Frontstr(6,Hole_ID,CordX,CordY),

```

```

Assert(temp_lr_layers_content_of_hole(No,CordX,CordY,Begin,End,Layers)),
VNo=No+1,
    Find_layers_content_and_depth_of_holes(VNo,Holes_ID).

```

Create_layerpairs:-

```
temp_lr_layers_content_of_hole(No,_,_,_,Left_Hole),  
Next_temp_layers_content_of_hole(No,Right_Hole),  
Create_a_layerpair(Left_Hole,_,Right_Hole,_,LayerPair,_,_),  
Assert(layerpairs_between_two_holes(No,LayerPair)),
```

Fail.

Create_layerpairs.

Next_temp_layers_content_of_hole(No,Hole):-

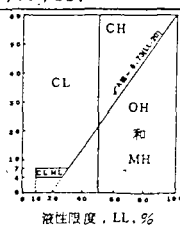
NextNo = No + 1,

```
temp_lr_layers_content_of_hole(NextNo,_,_,_,Hole),!
```

附錄五 統一土壤分類新表

(李成亨 1986, 12)

主要區分	次要區分	分類符號	其他指數條件	條件編號		
顆粒土	礫	GW	-	$Cu = \frac{D_{60}}{D_{10}} > 4$ 且 $Cc = \frac{(D_{30})^3}{D_{10} \times D_{60}} = 1 \sim 3$	G1	
		GP	-	本類土類中不符合條件G1者	G2	
	砂	GW-GM	-	G1 + G3	-	
		GW-GM-GC	-	G1 + G4	-	
		GW-GC	-	G1 + G5	-	
		GP-GM	-	G2 + G3	-	
		GP-GM-GC	-	G2 + G4	-	
		GP-GC	-	G2 + G5	-	
		粉土	GM ^d	LL ≤ 28 且 PI ≤ 6	阿太界限度低於 A 線或 PI < 4	G3
			GM ^u	LL > 28	阿太界限度高於 A 線且 PI = 4 ~ 7	G4
		GC	阿太界限度高於 A 線且 PI > 7	阿太界限度高於 A 線且 PI > 7	G5	
			$\begin{matrix} > 50\% \rightarrow < 5\% & 5\% \sim 12\% & > 12\% \\ > 4\# \# \# \# & < 200\# \# \# \# & > 12\% \\ < 50\% \rightarrow < 5\% & 5\% \sim 12\% & > 12\% \end{matrix}$	如果土壤之工程指數值正好在分界線上 則同時使用分界線上下之分類符號		
	黏土	砂	SW	-	$Cu > 6$ 且 $Cc = 1 \sim 3$	S1
			SP	-	本類土類中不符合條件S1者	S2
		SW-SM	-	S1 + S3	-	
SW-SM-SC		-	S1 + S4	-		
SW-SC		-	S1 + S5	-		
SP-SM		-	S2 + S3	-		
SP-SM-SC		-	S2 + S4	-		
SP-SC		-	S2 + S5	-		
粉土		SM ^d	LL ≤ 28 且 PI ≤ 6	阿太界限度低於 A 線或 PI < 4	S3	
		SM ^u	LL > 28	阿太界限度高於 A 線且 PI = 4 ~ 7	S4	
	SC	阿太界限度高於 A 線且 PI > 7	阿太界限度高於 A 線且 PI > 7	S5		
說明	> 50%					
說明	> 200# # # #	4# # # #	200# # # #	$Cu ; Cc ; PI ; LL$		
說明	< 50%	低塑性				
黏土類	低塑性	ML				
		CL				
		CL - ML				
		OL				
		OL				
	高塑性	MH				
		CH				
		OH				
		OH				
		有機土	Pt		腐殖土或其他含有機質之土壤	



附錄六 鑽孔資料表

A表

計劃名稱：台北盆地(南區)非黏性土壤之動態性質研究
 鑽孔編號：H1 X座標：304620 鑽孔傾角：90°
 提供單位：技術學院,營建系 Y座標：2768275 鑽孔孔徑：cm
 鑽探公司：全功 Z座標：9.6 鑽孔深度：30.85 m
 試驗公司：技術學院,營建系 水位計：
 鑽探日期：12-28-1988 埋設深度：
 鑽孔位置：大安國小

地下水位深度：2.85
 日期：12-28-88,

深度 (m)	土/岩層說明	土壤				採 樣 編 號	礫 石 %	砂 土 %	粉 土 %	粘 土 %	自 然 含 水 量 %	液性 限度 %	塑性 指數	總單 位重 t/ m ³	孔 隙 比 e
		岩石 分類	取 樣 率 %	RQD %	N 值										
0.00	深灰色砂質粉土														
1.30		CL			7 S001	0	3	59	38	21.5	31.5	24.3	1.91	2.66	
2.80		CL			5 S002	0	6	60	34	29.5	32.31	25.0	1.92	2.65	
3.70					T001							0			
4.30		CL			1 S003	0	2	65	33	31.8	30.6	25.0	1.82	2.63	
5.10					T002	0	10	83	7						
5.80	灰色粉土質細砂	SM			11 S004	0	73	16	11	14.6	-	NP	2.06	2.68	
7.30					33 S005					7.7			2.17		
8.80		SM			17 S006	38	44	10	8	9.8	---	NP	2.26	2.66	
10.30					27 S007										
11.80		SM			9 S008	0	76	14	10	15.7	---	NP	2.12	2.68	
12.60															
12.65	灰色粉土質黏土, 含有				T003										
13.30	機物	CL			7 S009	0	1	77	22	35.7	31.3	25.0	1.83	2.65	
14.15					T004										
14.80		CL			3 S010	0	2	55	43	39.0	39.4	25.6	1.78	2.67	
15.65					T005										
16.30		CL			2 S011	0	2	47	51	35.7	30.3	23.5	1.81	2.68	
17.80					3 S012										
19.30		CL			4 S013	0	4	66	30	37.8	36.2	23.1	1.85	2.59	
20.10					T006										
20.80		CL			3 S014	0	2	62	36	34.6	35.6	25.0	1.80	2.63	
21.80					T007										
22.30		CL			15 S015	0	34	43	23	24.9	32.6	20.0	2.00	2.62	
23.55					T008										
24.20		CL			16 S016	0	30	44	26	24.4	37.2	22.6	1.94	2.65	
25.55	深灰色粉土質細砂				T009										
26.20		SM			28 S017	0	55	25	20	19.4			2.04	2.62	
27.60					T010										
28.10	深灰色粉土質黏土	CL			30 S018	0	19	36	45	25.7	43.6	26.7	1.95	2.57	
29.00															
29.30	棕色砂質黏土, 含風化 碎岩屑	SC			34 S019	0	56	25	19	28.3	38.2	23.7	1.94	2.63	
29.50															
30.70	卵礫石砂	GM		>90	5020	3	48	24	25	33.6	37.1	21.6	2.01	2.64	
30.85															

謝 誌

隨著重大土木工程之興建和天災人禍之發生，足以緊急處理地下事件之「大地工程資料」逐漸成為大眾關懷之基本資料。內政部建築研究所籌備處和本研究群有幸於民國七十六年即著手規劃「大地工程資料庫系統」和電腦化之研究工作，並且分年建檔，逐步邁向實用性之目標。本研究群擬向關心且推動本研究案之下列人士致最大謝意：

內政部營建署	署長	蔡兆陽先生
	副署長	張世典先生
	組長	張德周先生
	科長	李玉生先生

內政部建築研究所籌備處	副主任	蕭江碧先生
	副研究員	毛 犖先生

摘 要

由於地層的土壤成份直接地影響營建工程之方法，因此若事前有效地提供地層資訊，則能協助營建工程之規劃與設計。本研究即試圖以專家系統之方法，搜集大地工程之地質資料，並模仿專家之推理程序，以建構一土層剖面圖專家系統，協助工程之設計規劃。

本研究取消 Aho與Peterson之字串辨識理論中之「取代錯換」功能，但保留符合土壤分層特性之其餘錯換定義，並引用Fu之權重把念，作為分層辨識之方法；更改圖形理論之最短路徑求解過程作為土層匹配之路徑搜尋程序；以大中取小原則，作為土層是否再細分之基礎。使用邏輯語言Turbo Prolog及其內建推論機，並以自創之土壤分層辨識方法另造一外建推論機，共同進行土壤分層之推論工作。外建推論機之推論步驟可分為鄰孔之比對，層次細分，多孔層次之最佳匹配，以及土層劃分之決定等四大步驟。

土層剖面圖專家系統即以上述之外建推理機為基礎，配合專家所提供之土壤對等值之領域知識，及土層分層及匹配之決策等知識建構而成。本系統之組成包括使用者介面、推理機、工作記憶區、知識庫、知識編輯模組、外部資料轉換模組。

目 錄

謝誌	I
摘要	II
一、前言	1
二、專家系統	3
2.1 專家系統之定義	3
2.2 專家系統之分類	4
2.3 專家系統之架構	5
2.4 知識表示	8
2.5 推理機之控制策略	9
2.6 專家系統的生命週期	11
三、土層剖面圖	15
3.1 鑽探計劃	15
3.2 由地質探查資料繪製地質剖面圖	16
3.3 土壤分類	16
3.4 探勘記錄解釋之專家系統	19
四、土層剖面圖專家系統推論機之設計	22
4.1 推理機設計策略之研究	22
4.2 人工智慧中相關之匹配理論	28
4.3 鑽探基本資料之表達	33
4.4 土岩對等值與層次相似值之介紹	34
4.5 推理機之規則	40
五、土壤分層專家系統之功能	50
5.1 工具選擇	50
5.2 整體架構	51
5.3 系統功能與賡作	55
六、結論與建議	65
6.1 結論	65
6.2 建議	65
附錄一 土壤對等值表	67
附錄二 N 值對等值表	68
附錄三 土壤對等值與 N 值對等值	69
附錄四 土層剖面圖專家系統推理機	78
附錄五 統一土壤分類新表	91
附錄六 鑽孔資料表	92